ON THE IMPLEMENTATION OF REDUCED SUB-OPTIMAL KALMAN FILTERS, FOR DISCRETE, LINEAR, STOCHASTIC PROCESSES WITH TIME-INVARIANT DYNAMICS

by

Juan Francisco Lara

United States
Naval Postgraduate School

THESIS

ON THE IMPLEMENTATION OF REDUCED SUB-OPTIMAL KALMAN FILTERS, FOR DISCRETE, LINEAR, STOCHASTIC PROCESSES WITH TIME-INVARIANT DYNAMICS

by

Juan Francisco Lara

December 1969

On the Implementation of Reduced, Sub-Optimal Kalman Filters, for Discrete, Linear, Stochastic Processes with Time-Invariant Dynamics

by

Juan Francisco Lara
Lieutenant Commander, Venezuelan Navy
B.S., Venezuelan Naval Academy, 1956

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1969

ABSTRACT

Three different approaches to the problem of implementing a reduced-order, sub-optimal Kalman filter for a discrete, linear stochastic process, with time-invariant dynamics, are presented.

A first method, A, is based upon the partitioning of the system dynamics. A second method, B, is implemented using matrix pseudo-inversion and a third method, C, is based upon reduction of the original process to one of lower order using the dominant roots of the system. An expression for the performance degradation in method A is derived. In method B, expressions for the sub-optimal estimation error, and sub-optimal variance of estimation error are derived.

The several methods are applied to a fourth-order process for illustration.

TABLE OF CONTENTS

ACKNOWLEDGEMENT

The author wishes to express his deepest gratitude to Dr. James S. Demetry for his most valuable guidance and patience in this work. He also wishes to thank his wife, Aida, for her constant encouragement and understanding.

I. INTRODUCTION

In any modern control problem involving filtering of discrete or continuous stochastic linear processes, it is clear that the computational requirements associated with the implementation of a Kalman filter increase considerably as the dimension of the system increases. A simpler (reduced) configuration of the filter with sub-optimal performance may be acceptable, mainly if some preselected r states (where $r \leq n$, and n represents the order of the original system) are of primary interest in the estimation process and if the performance degradation incurred when using the reduced model is not severe.

With this idea of implementing a reduced, discrete, sub-optimal filter, based upon the optimal Kalman estimation theory (Ref.1.), a library search was done, yielding two papers of particular interest which present different approaches to the problem of reduced estimation for continuous dynamic processes. Meditch (Ref.3) in 1964, developed a theory of sub-optimal filtering for continuous dynamic processes, in which the process vector is divided by means of matrix pseudo inversion, with separate filters designed for each part. The sub-optimal estimates are obtained by combining the several filter outputs. As Meditch points out in his paper, an important feature of his continuous formulation is that it provides a powerful tool for preliminary analysis of sub-optimal filters.

Huddle and Wismer (Ref.4) in 1968 based a sub-optimal filter design on a "primary system", which is implemented by dividing the original process into two separate systems, using partitioning of the continuous dynamics of the system, and neglecting the remaining "secondary system".

Another successful technique for simplifying filter computations has been investigated by Aoki and Huddle (Ref.5), wherein a procedure is developed to estimate the state vector of a discrete stochastic system, when constraints are imposed on the number of memory elements of the estimator.

This thesis presents both an extension of two of the methods referred to above (Ref.3 and Ref.4) to the discrete case, and a third approach to the same problem based upon the use of the dominant roots of the full process.

The study of the three different approaches to the reduced sub-optimal filter problem is presented in a coherent form as follows. Chapter II describes the general discrete linear stochastic process. In Chapter III, the work of Huddle and Wismer (Ref.4) is extended to the discrete case, and the performance degradation of the reduced filter is obtained. An example using a fourth-order process is solved estimating only two states of the full system, in order to illustrate the method. In Chapter IV, the Meditch (Ref.3) reduced filter is adapted to the discrete case, and recursive equations for the sub-optimal estimation error and sub-optimal covariance of

estimation error are derived and used as a means of comparing the sub-optimal with optimal performances. The same fourth-order system used in Chapter III, is used here as an example, and again, only two states are estimated. In Chapter V, another reduced filter is implemented, using the dominant roots of the complete system in order to produce the dynamics of the reduced filter. Once more, the technique is illustrated using the original plant of the previous examples.

In the three examples presented as illustrations, the same complete process is used in order to compare the results of the three different methods. In each one of the examples, a Monte Carlo simulation of the full process was performed, and optimal Kalman estimation of the whole state vector was accomplished in order to establish the performance degradation of the reduced filters.

Appendix A presents the basic principles of matrix pseudo-inversion; more complete treatments of this subject appear in Penrose (Ref.8) and Werther (Ref.9).

Appendix B and C include the computer programs used in solving the examples. These programs were not intended to be computationally efficient either in time, or in storage.

II. THE DISCRETE (DISCRETIZED) LINEAR STOCHASTIC PROCESS WITH TIME-INVARIANT DYNAMICS

A linear, continuous stochastic dynamic process may be described in general (Ref.7) by the vector matrix differential equation

$$\dot{\underline{x}}(t) = A(t)\underline{x}(t) + B(t)\left[\underline{u}(t)+\underline{w}(t)\right] \tag{2.1}$$

with output

$$\underline{y}(t) = H(t)\underline{x}(t) \tag{2.2}$$

where: $\underline{x}(t)$ is a (n x 1) vector, representing the state variables of the system

$\underline{u}(t)$ is a (m x 1) vector (m n), representing the deterministic input to the system

$\underline{w}(t)$ is a (m x 1) vector representing the stochastic input to the system

$A(t)$ is a (n x n) system matrix

$B(t)$ is a (m x n) distribution matrix

$\underline{y}(t)$ is a (p x 1) vector of system outputs

$H(t)$ is a (p x n) measurement matrix.

It will be assumed that measurement of the output can be achieved only with the addition of noise, so the measurement equation is defined as

$$\underline{z}(t) = \underline{y}(t) + \underline{v}(t), \qquad (2.3)$$

where $\underline{v}(t)$ is a (p x 1) vector representing the stochastic measurement noise and $\underline{z}(t)$ is a (p x 1) vector of measured outputs.

The process is visualized better by means of Fig.2.1, where the double lines indicate a signal flow characteristic of all matrix block diagrams.

It is assumed for simplicity that the process is stationary (time-invariant); that is, all the elements of the A(t), B(t) and H(t) matrices are constants. It is further assumed that the process is to be sampled at discrete times and that $\underline{u}(t)$ and $\underline{w}(t)$ are constant vectors during each sampling period. The general solution for Eq. (2.1) under these conditions is expressed as

$$\underline{x}(t) = \epsilon^{At}\underline{x}(t_o) + \epsilon^{At}\int_o^t \epsilon^{-A\tau} B\left[\underline{u} + \underline{w}\right] d\tau \ . \qquad (2.4)$$

If the sampling time is assumed to be T seconds, Eq. (2.4) may be rewritten as

$$\underline{x}(T) = \epsilon^{AT}\underline{x}(t_o) + \epsilon^{AT}\int_o^T \epsilon^{-A\tau} d\tau \ B\left[\underline{u} + \underline{w}\right] \ . \qquad (2.5)$$

Defining

$$\underline{\Phi}(T) \triangleq \epsilon^{AT} \ , \qquad (2.6a)$$

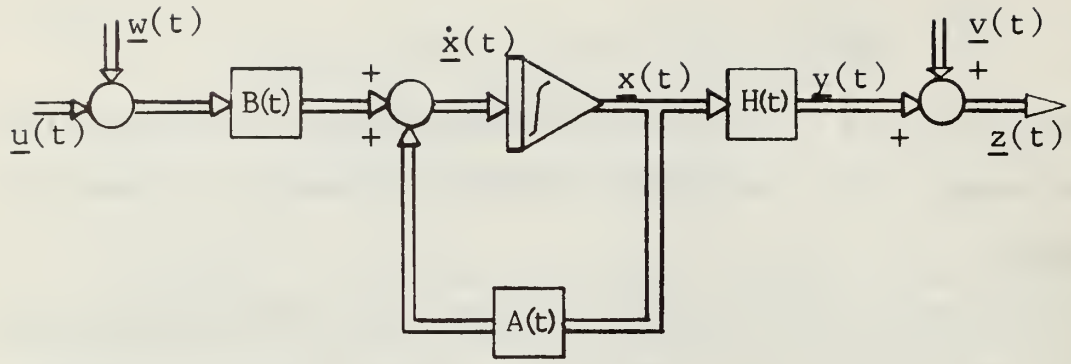


Fig. 2.1 - Block diagram for a continuous stochastic process with time-varying dynamics.

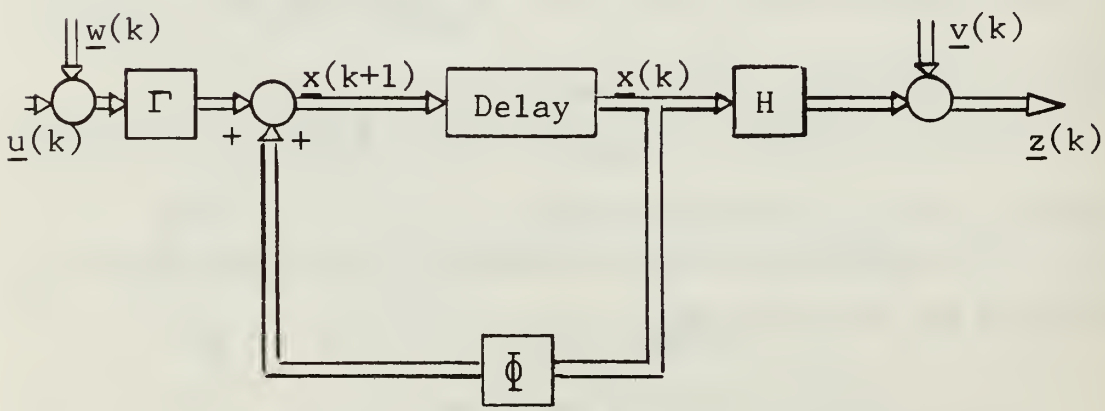


Fig. 2.2 - Block diagram for a discrete stochastic process with time-invariant dynamics.

and

$$\Gamma(T) \triangleq e^{AT} \int_0^T e^{-A\tau} d\tau \, B \ , \tag{2.6b}$$

where $\Phi(T)$ (n x n) and $\Gamma(T)$ (n x m) are called the discrete state transition and distribution matrices, respectively. The discretized[1] form of Eq. (2.1) becomes in the stationary case

$$\underline{x}(k+1) = \Phi(T)\underline{x}(k) + \Gamma(T)\left[\underline{u}(k) + \underline{w}(k)\right] \ . \tag{2.7}$$

$\Phi(T)$ and $\Gamma(T)$ may be obtained by application of the Laplace transformation, i.e.,

$$\Phi(T) = \mathcal{L}^{-1}\left[(sI - A)^{-1}\right]\Big|_{t=T} \ , \tag{2.8a}$$

1

Distinction is made between a truly discrete system, and one which is continuously operating but driven and measured at discrete times. The linear difference equation for the truly discrete system will be

$$\underline{x}(k+1) = A_D\underline{x}(k) + B_D\,\underline{u}(k) + \underline{w}(k) \ .$$

The terminology "discrete", "discretized" is used without distinction from this point on.

$$\Gamma(T) = \mathcal{L}^{-1}\left[\frac{\Phi(s)B}{s}\right]_{t=T} . \qquad (2.8b)$$

The discretized, linear stochastic process with time-invariant coefficients will hereafter be represented by the model (see Fig. 2.2)

$$\underline{x}(k+1) = \Phi\underline{x}(k) + \Gamma\left[\underline{u}(k) + \underline{w}(k)\right] \qquad (2.9)$$

with measurement equation

$$\underline{z}(k) = H\underline{x}(k) + \underline{v}(k) . \qquad (2.10)$$

Eqs. (2.9) and (2.10) will be used throughout this thesis as the model of the complete discrete system for which reduced sub-optimal filters are implemented. Furthermore, $\underline{w}(k)$ and $\underline{v}(k)$ will be considered to be vectors of independent, gaussian random processes (white noise) with zero means and fully described by the following statistics:

$$E\left[\underline{w}(k)\right] = \underline{0} \quad \text{for all } k , \qquad (2.11)$$

$$E\left[\underline{w}(k)\cdot\underline{w}^T(i)\right] = E\left[\underline{w}(k)\right]\cdot E\left[\underline{w}^T(i)\right] = 0 \quad k \neq i, \qquad (2.12)$$

$$E\left[\underline{w}(k)\cdot\underline{w}^T(k)\right] = \text{known quantity for all } k , \qquad (2.13)$$

$$E\left[\underline{v}(k)\right] = \underline{0} \qquad \text{for all } k\ , \tag{2.14}$$

$$E\left[\underline{v}(k)\cdot\underline{v}^T(i)\right] = E\left[\underline{v}(k)\right]\cdot E\left[\underline{v}^T(i)\right] = 0 \quad k \neq i, \tag{2.15}$$

and

$$E\left[\underline{v}(k)\cdot\underline{v}^T(k)\right] = R \quad \text{for all } k\ , \tag{2.16}$$

where R is a (p x p) matrix of covariance of measurement noise.

III. IMPLEMENTATION OF A REDUCED DISCRETE, LINEAR KALMAN FILTER BY PARTITIONING THE DYNAMICS OF THE ORIGINAL PROCESS

This chapter presents a discrete solution to the problem of implementing a sub-optimal filter by means of partitioning the original discrete system (Fig. 2.2) into two reduced models, a "primary system" model which will include only a selected subset of the state vector components whose estimation is of primary interest, and a "secondary system" which will be formed by the remaining state varibales. Then, the primary plant will have reduced dimensions and the implementation of a reduced Kalman filter based upon this primary model will be possible.

A. DERIVATION OF THE REDUCED FILTER EQUATIONS

It is assumed that the system dynamics are represented by Eq. (2.9) and that the statistics of the forcing and measurement noise are fully described by Eqs. (2.11) through (2.16). For simplicity it will be further assumed that there is no deterministic forcing function $\underline{u}(k)$. The stochastic process may then be expressed as

$$\underline{x}(k+1) = \Phi\underline{x}(k) + \Gamma\,\underline{w}(k) \ , \qquad (3.1)$$

with measurement

$$\underline{z}(k) = H\underline{x}(k) + \underline{v}(k).$$

By matrix partitioning, the discrete dynamics of the full process may be rewritten as

$$\begin{bmatrix} \underline{x}_1(k+1) \\ \hline \underline{x}_2(k+1) \end{bmatrix} = \begin{bmatrix} \Phi_1 & \Phi_{c1} \\ \hline \Phi_{c2} & \Phi_2 \end{bmatrix} \begin{bmatrix} \underline{x}_1(k) \\ \hline \underline{x}_2(k) \end{bmatrix} + \begin{bmatrix} \Gamma_1 & 0 \\ \hline 0 & \Gamma_2 \end{bmatrix} \begin{bmatrix} \underline{w}_1(k) \\ \hline \underline{w}_2(k) \end{bmatrix}. \quad (3.2)$$

With the full state vector being an (n x 1) matrix, then:

$\underline{x}_1$ represents an (n_1 x 1) state vector which includes all those states of particular interest in the estimation process;

$\underline{x}_2$ represents an (n_2 x 1) vector of the remaining state vector components ($n_1 + n_2 = n$);

Φ_1 is an (n_1 x n_1) matrix, representing the dynamics of $\underline{x}_1$;

Φ_2 is an (n_2 x n_2) matrix, representing the dynamics of $\underline{x}_2$;

Φ_{c1} and Φ_{c2} are (n_1 x n_2) and (n_2 x n_1) matrices respectively, indicating the coupling between the primary and secondary systems;

$\underline{w}_1$ is a (m_1 x 1) vector of forcing (white noise) functions;

$\underline{w}_2$ is a (m_2 x 1) vector of forcing (white noise) functions with $m_1 + m_2 = m$; and

Γ_1 and Γ_2 are (n_1 x m_1), (n_2 x m_2) distribution matrices respectively.

The observation matrix H is partitioned as follows,

$$H = \left[H_1 \mid H_2 \right] .$$

The measurement equation may therefore be rewritten as

$$\underline{z}(k) = \left[H_1 \mid H_2 \right] \begin{bmatrix} \underline{x}_1(k) \\ \hline \underline{x}_2(k) \end{bmatrix} + \underline{v}(k) . \qquad (3.3)$$

It is assumed that all the measured outputs are included in the primary system, which makes $H_2 = 0$. This partitioning may not always be possible, and even if it is, it may require special rearrangement of the $\underline{z}$ vector.

In Eq. (3.3) $\underline{z}(k)$ is the original $(p \times 1)$ measured vector and H_1 is the $(p \times n_1)$ measurement matrix for the primary system. It may be seen from this special partitioning of the H matrix that the primary system vector $\underline{x}_1$ will not only be formed by those states of primary interest in the estimation process, but will also include those states whose estimation may not be needed, but whose outputs are being measured. This will establish the condition that $n_1 \geq p$.

As a result of the preceeding discussion, Eq. (3.3) becomes

$$\underline{z}(k) = H_1 \underline{x}_1(k) + \underline{v}(k) , \qquad (3.4)$$

which represents the measurement equation for the reduced system.

The general system may be described as consisting of a primary system.

$$\underline{x}_1(k+1) = \Phi_1\underline{x}_1(k) + \Phi_{c1}\underline{x}_2(k) + \Gamma_1\underline{w}_1(k) \ , \qquad (3.5a)$$

and a secondary system described by,

$$\underline{x}_2(k+1) = \Phi_{c2}\underline{x}_1(k) + \Phi_2\underline{x}_2(k) + \Gamma_2\underline{w}_2(k) \ . \qquad (3.5b)$$

Fig. (3.1) shows a complete representation of the partitioned system.

In order to reduce computational requirements in the estimation of the preselected states, the design of the filter is based on the dynamics of the primary system, which can be represented as a simplified model of the reduced process by

$$\bar{\underline{x}}_1(k+1) = \Phi_1\underline{x}_1(k) + \Gamma_1\underline{w}_1(k) \ , \qquad (3.6)$$

[1]

---

[1] A bar over a quantity indicates a reduced-model quantity.

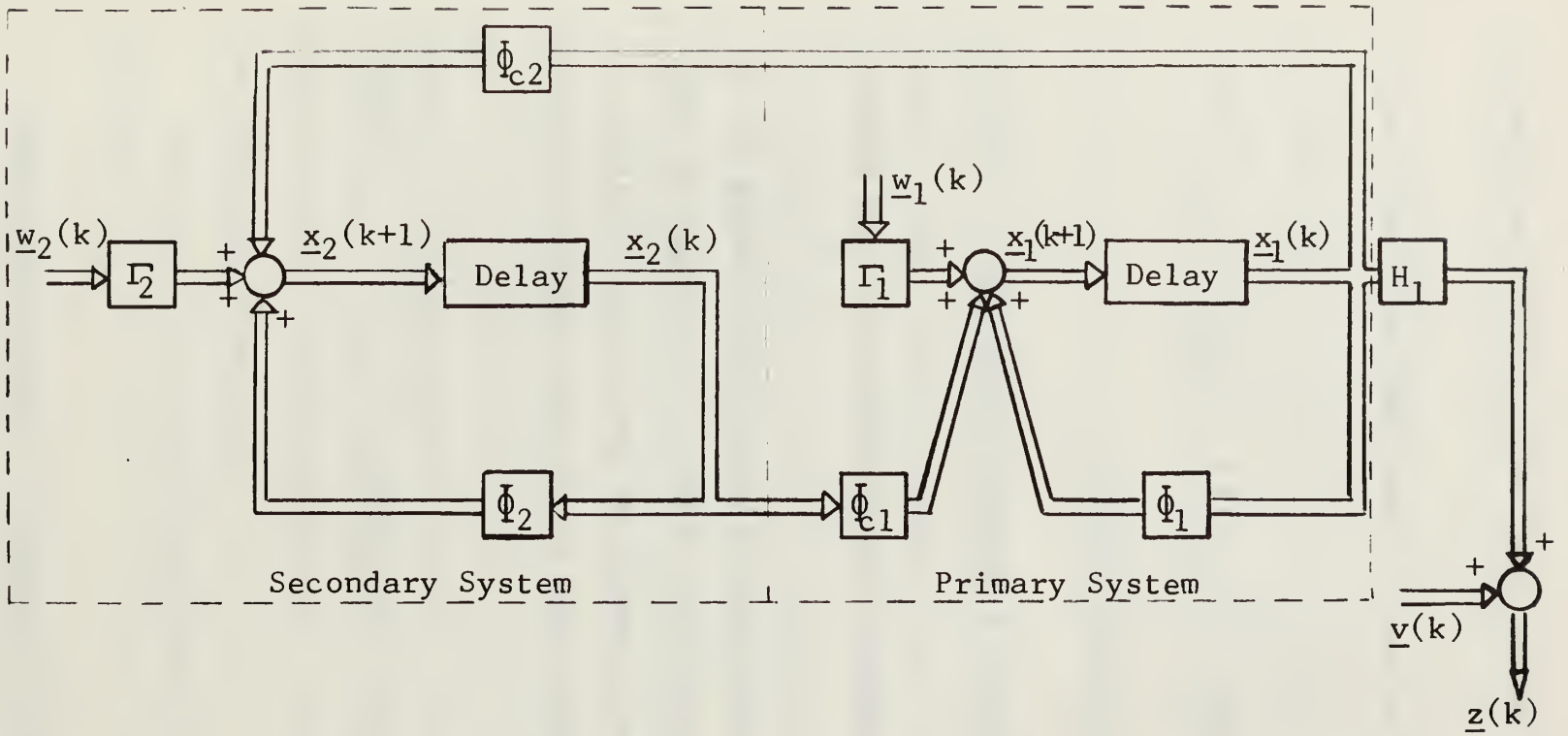


Fig. 3.1 - Block diagram representation of the complete system partitioned into a primary and a secondary systems. Measurements are obtained only from the primary system. The deterministic component of the forcing function vector is not considered.

with measurement

$$\underline{z}(k) = H_1 \underline{x}_1(k) + \underline{v}(k) \ .$$

In the simplified model of the primary system the term involving $\Phi_{c1}\underline{x}_2(k)$ has been neglected due to the fact that the secondary state vector $\underline{x}_2(k)$ is not considered at all in the implementation of the reduced filter. The consequences of using Eq. (3.6) instead of Eq. (2.5a), will be analyzed in Section B of this chapter.

If the Kalman filter algorithm is applied using the dynamics of the full plant, the matrix equation for the gain G(k), the covariance of estimation error P(k/k) and the prediction covariance P(k+1/k) are given in the discrete case by

$$G(k) = P(k/k-1)H^T \left[H\, P(k/k-1)H^T + R \right]^{-1} , \qquad (3.7a)$$

$$P(k/k) = \left[I - G(k)H \right] P(k/k-1) , \qquad (3.7b)$$

and

$$P(k+1/k) = \Phi\, P(k/k)\Phi^T + Q , \qquad (3.7c)$$

where

$$Q = \Gamma\, E\left[\underline{w}(k)\underline{w}^T(k)\right]\Gamma^T \ .$$

G(k) is an (n x p) matrix, P(k/k), P(k/k-1) and P(k+1/k) are (n x n)matrices.

The estimation equation is given by

$$\underline{\hat{x}}(k/k) = \underline{\Phi}\underline{\hat{x}}(k-1/k-1) + G(k)\left[\underline{z}(k) - H\underline{\Phi}\underline{\hat{x}}(k-1/k-1)\right] \tag{3.8}$$

where $\underline{\hat{x}}(k/k)$ is an (n x 1) vector of estimated states.

By induction, the reduced sub-optimal expressions using the dynamics of Eq. (3.6) are

$$G_1(k) = P_1(k/k-1)H_1^T\left[H_1P_1(k/k-1)H_1^T + R\right]^{-1} , \tag{3.9a}$$

$$P_1(k/k) = \left[I - G_1(k)H_1\right]P_1(k/k-1) , \tag{3.9b}$$

and similarly

$$P_1(k+1/k) = \underline{\Phi}_1P_1(k/k)\underline{\Phi}_1^T + Q_1 , \tag{3.9c}$$

where

$$Q_1 = \Gamma_1E\left[\underline{w}_1(k)\underline{w}_1^T(k)\right]\Gamma_1^T .$$

$P_1(k/k)$ (n_1 x n_1) represents the sub-optimal covariance of estimation error and $P_1(k+1/k)$ (n_1 x n_1) is the sub-optimal prediction covariance of estimation error.

The sub-optimal estimation equation is

$$\hat{\underline{x}}_1(k/k) = \Phi_1 \hat{\underline{x}}_1(k\text{-}1/k\text{-}1) + G_1(k)\left[\underline{z}(k) - H_1 \Phi_1 \hat{\underline{x}}_1(k\text{-}1/k\text{-}1)\right]. \tag{3.10}$$

Considering the problem of initialization for the discrete reduced covariance equations, it is seen as a good approach to use for $P_1(0/-1)$, the partitioned portion of $P(0/-1)$ corresponding to $P_{11}(0/-1)$, or considering that

$$P(0/-1) = \left[\begin{array}{c|c} P_{11}(0/-1) & P_{12}(0/-1) \\ \hline P_{21}(0/-1) & P_{22}(0/-1) \end{array}\right]$$

then

$$P_1(0/-1) = P_{11}(0/-1) .$$

For the filter initial states,

$$\hat{\underline{x}}_1(0/-1) = \underline{\hat{x}}_1(0/-1) ,$$

where

$$\hat{\underline{x}}(0/-1) = \left[\begin{array}{c} \underline{\hat{x}}_1(0/-1) \\ \hline \underline{\hat{x}}_2(0/-1) \end{array}\right] .$$

Fig. 3.2 shows a complete block diagram of the reduced plant and filter.

B. REDUCED FILTER DEGRADATION

Because the dynamics of a simplified system represented by

$$\underline{\bar{x}}_1(k+1) = \bar{\Phi}_1\underline{x}_1(k) + \bar{\Gamma}_1\underline{w}_1(k) \tag{3.6}$$

were used for the implementation of the reduced filter, instead of the dynamics of the primary system

$$\underline{x}_1(k+1) = \bar{\Phi}_1\underline{x}_1(k) + \bar{\Phi}_{c1}\underline{x}_2(k) + \bar{\Gamma}_1\underline{w}_1(k) \ , \tag{3.5a}$$

the estimation of $\underline{x}_1$ is caused to be sub-optimal and in error. This error may be considered the performance degradation of the sub-optimal filter.

Considering the reduced filter, its estimation error covariance is given by

$$V(k) = E\left[\underline{x}_1(k) - \underline{\bar{x}}_1(k/k)\right]^2 \tag{3.11}$$

where

$$\underline{x}_1(k) = \bar{\Phi}_1\underline{x}_1(k-1) + \bar{\Phi}_{c1}\underline{x}_2(k-1) + \bar{\Gamma}_1\underline{w}_1(k-1) \ , \tag{3.5a}$$

and by proper use of Eq. (3.6) it may be rewritten as

$$\underline{x}_1(k) = \underline{\bar{x}}_1(k) + \bar{\Phi}_{c1}\underline{x}_2(k-1) \ . \tag{3.12}$$

In the same way,

$$\bar{\underline{x}}_1(k/k) = \Phi_1\hat{\underline{x}}_1(k-1/k-1)+G_1(k)\left[\underline{z}(k) - H_1\Phi_1\hat{\underline{x}}_1(k-1/k-1)\right] , \tag{3.10}$$

which by substitution of Eq. (3.4) for $\underline{z}(k)$, gives

$$\bar{\underline{x}}_1(k/k) = \Phi_1\hat{\underline{x}}_1(k-1/k-1)+ G_1(k)\left[H_1\underline{x}_1(k) +\underline{v}(k) - H_1\Phi_1\hat{\underline{x}}_1(k-1/k-1)\right] .$$

Further substitution of Eq. (3.12) for $\underline{x}_1(k)$ gives

$$\hat{\underline{x}}_1(k/k) = \Phi_1\hat{\underline{x}}_1(k-1/k-1) + G_1(k)H_1\left[\bar{\underline{x}}_1(k) + \Phi_{c1}\underline{x}_2(k-1) - \Phi_1\hat{\underline{x}}_1(k-1/k-1)\right] + G_1(k)\underline{v}(k) . \tag{3.13}$$

Using Eqs. (3.12) and (3.13), the experimental estimation error covariance V(k) (Eq. 3.11) may be rewritten, after proper rearrangement, as

$$V(k) = E\left[\left[(I_1 - G_1(k)H_1)(\bar{\underline{x}}_1(k) - \Phi_1\hat{\underline{x}}_1(k-1/k-1) + (I_1 - G_1(k)H_1)\Phi_{c1}\underline{x}_2(k-1) + G_1(k)\underline{v}(k)\right]^2\right]. \tag{3.14}$$

By application of matrix algebra and elemental algebra of expectations, Eq. (3.14) may be expanded and after some manipulation gives

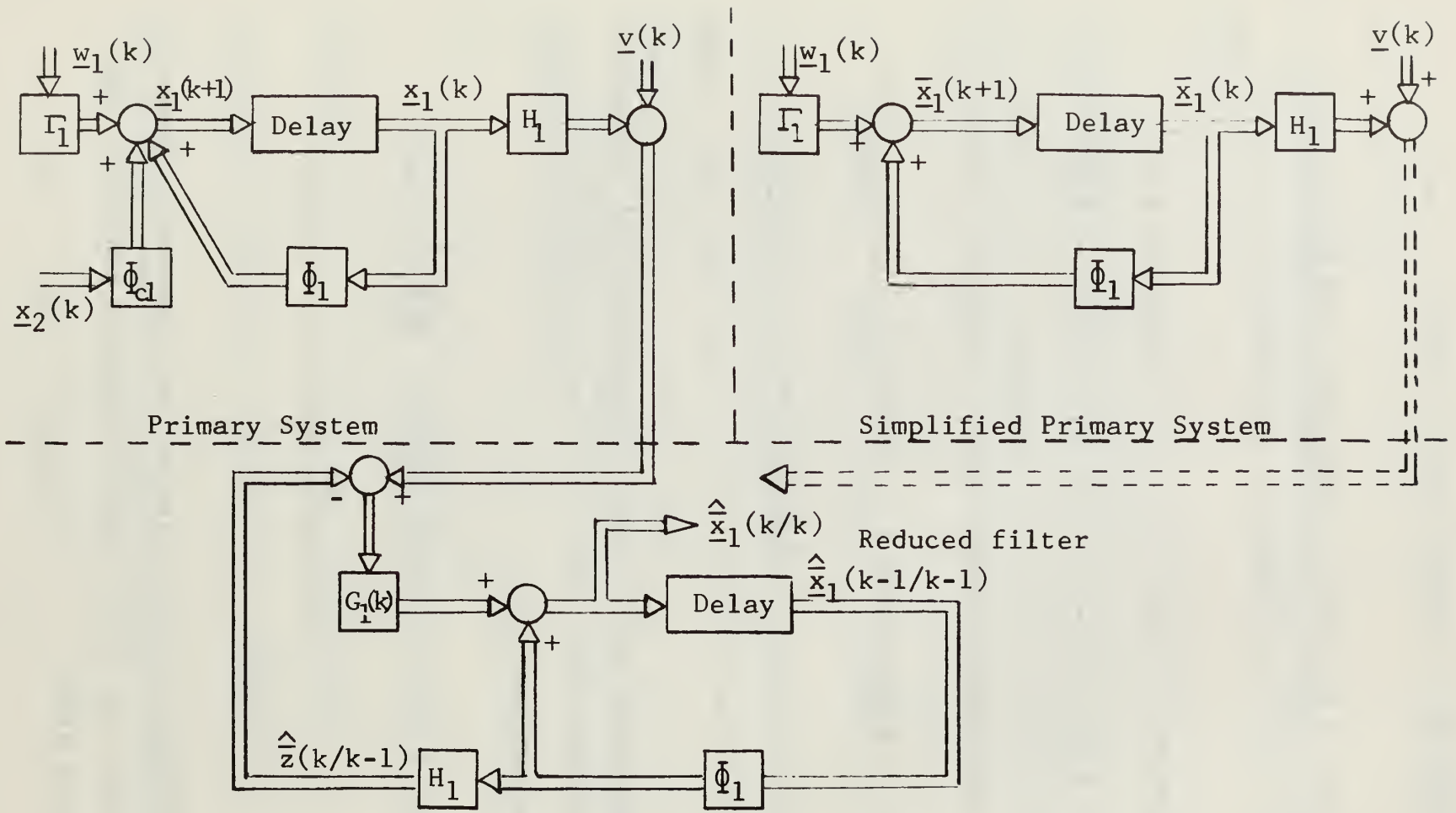


Fig. 3.2 - Block diagram representation of the reduced filter obtained by partitioning of the complete system dynamics.

$$V(k) = \left[I_1 - G_1(k)H_1\right] P_1(k/k-1) + \left[I_1 - G_1(k)H_1\right] \Big[\Phi_{c1} E\left[\underline{x}_2^2(k-1)\right] \Phi_{c1}^T + E\left[\underline{\bar{e}}_1(k/k-1)\underline{x}_2^T(k-1)\right] \Phi_{c1}^T + \Phi_{c1} E\left[\underline{x}_2(k-1)\underline{\bar{e}}_1^T(k/k-1)\right] \Big] \left[I_1 - G_1(k)H_1\right]^T , \quad (3.15)$$

where

$$\underline{\bar{e}}_1(k/k-1) = \underline{\bar{x}}_1(k) - \underline{\hat{\bar{x}}}_1(k-1/k-1) .$$

The first term on the right hand side of Eq. (3.15) can be easily recognized as $P_1(k/k)$, the sub-optimal theoretical variance of estimation error based on the reduced model. It is intuitively expected that $P_1(k/k) > P_{11}(k/k)$. The remaining terms of Eq. (3.15) represent the further degradation of the reduced filter and are seen to be functions of that partitioned portion of Φ which couples $\underline{x}_2$ to the primary system, i.e., Φ_{c1}. This suggests that as the coupling between $\underline{x}_2$ and the primary system (Φ_{c1}) decreases, the quantity $P_1(k/k)$ more accurately represents the actual estimation error covariance.

C. EXAMPLE 1

The process used for this example was a fourth-order model, with continuous linear stationary dynamics subjected to a forcing function having both a deterministic and a random component, with the latter having zero mean and known variance. It was assumed that only one state variable was measured, subject to the presence of gaussian measurement noise with zero mean and known variance. It

was further assumed that both random sequences have statistics as described by Eqs. (2.11) through (2.16). The reduced filter was implemented in order to estimate only two states (x_1, x_2).

The fourth-order continuous system is described by

$$\dot{\underline{x}}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -400 & -320 & -118 & -19 \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 400 \end{bmatrix} \left[u(t)+w(t) \right] .$$

A deterministic function $u(t) = 0.5$ was used in the example. The random forcing function $w(t)$ was assumed to have zero mean and variance equal to 0.01 square units, i.e., $w(t) : N(0.0, 0.01)$.

The measurement equation is

$$z(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \underline{x}(t) + v(t) ,$$

with the measurement noise $v(t)$, considered to have zero mean and variance equal to 0.25 square units, i.e., $v(t) : N(0.0, 0.25)$.

Before applying the reduced-filter technique discussed in Section III, A, the continuous process was discretized using the ideas presented in Chapter II, so the discretized system could be described as

$$\underline{x}(k+1) = \Phi \underline{x}(k) + \Gamma \left[u(k) + w(k) \right] ,$$

with measurement

$$z(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \underline{x}(k) + v(k) ,$$

where

$$\Phi = e^{AT} \tag{2.6a}$$

was computed using the series expansion

$$\Phi = I + AT + \frac{1}{2!}(AT)^2 + \frac{1}{3!}(AT)^3 + \cdots\cdots,$$

and

$$\Gamma = e^{AT} \int_0^T e^{-A\tau} d\tau \; B , \tag{2.6b}$$

was computed by means of

$$I = (I + AT + \frac{1}{2!}(AT)^2 + \cdots)(IT - A\frac{T^2}{2!} + A^2\frac{T^3}{3!} - \ldots)B.$$

A Monte Carlo simulation[1] was performed using 200 replications of the trajectory, each trajectory being formed by 50 samples with sampling time T = 0.05 seconds.

---

[1] An IBM System/360-67 computer was used for solving all the examples presented in this thesis. A copy of the main computer program for this example is presented in Appendix B.

This simulation was carried out in order to produce approximate ensemble values for the experimental mean and experimental variance of estimation error in x_1 and x_2, as a function of time from t = 0, for both the complete, optimal filter and the reduced filter.

For the Monte Carlo simulation each state was assumed to have uncorrelated gaussian initial conditions with zero mean and the following variances:

$$\sigma^2 x_1 = 0.5 \text{ square units}$$

$$\sigma^2 x_2 = 2.0 \text{ square units}$$

$$\sigma^2 x_3 = 2.0 \text{ square units}$$

$$\sigma^2 x_4 = 4.0 \text{ square units.}$$

In order to obtain an unbiased estimate from the Kalman filter, the initial values assigned to the filter state variables were

$$\underline{\hat{x}}(0/-1) = E\left[\underline{x}(0)\right]$$

or

$$\underline{\hat{x}}(0/-1) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

and the initial value of P(0/-1) was

$$P(0/-1) = E\left[\left[\underline{x}(0) - \hat{\underline{x}}(0/-1)\right]\left[\underline{x}(0) - \hat{\underline{x}}(0/-1)\right]^T\right]$$

$$= E\left\{\left[\underline{x}(0) - E\left[\underline{x}(0)\right]\right]\left[\underline{x}(0) - E\left[\underline{x}(0)\right]\right]^T\right\}$$

$$= \mathrm{cov}\left[\underline{x}(0)\right] ,$$

or

$$P(0/-1) = \begin{bmatrix} 0.5 & \bigcirc & \\ & 2.0 & \\ \bigcirc & & 2.0 \\ & & 4.0 \end{bmatrix} .$$

The reduced filter was implemented to estimate only two states (x_1 and x_2), and for this purpose partitioning was applied as follows. The Φ and Γ matrices (from Table III-1)

$$\Phi(0.05) = \left[\begin{array}{cc|cc} 0.9990 & 0.0449 & 0.0012 & 0.0000 \\ -0.0066 & 0.9946 & 0.0480 & 0.0009 \\ \hline -0.3644 & -0.2981 & 0.8871 & 0.0307 \\ -12.2713 & -10.1815 & -3.9181 & 0.3043 \end{array}\right]$$

$$\Gamma^T(0.05) = \left[\begin{array}{cc|cc} 0.0001 & 0.0066 & 0.3644 & 12.2713 \end{array}\right],$$

were partitioned as shown to obtain

$$\Phi_1 = \begin{bmatrix} 0.9990 & 0.0449 \\ -0.0066 & 0.9946 \end{bmatrix},$$

and

$$\Gamma_1 = \begin{bmatrix} 0.0001 \\ 0.0066 \end{bmatrix},$$

Correspondingly,

$$P_1(0/-1) = \begin{bmatrix} 0.5 & 0 \\ 0 & 2.0 \end{bmatrix}$$

$$\hat{\underline{x}}_1(0/-1) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$H_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

and

$$Q_1 = \begin{bmatrix} 0.000 & 0.000 \\ 0.000 & 0.000 \end{bmatrix}. \quad \text{(See Table III-1)}$$

The reduced filter was then implemented using the set of Eqs. (3.9a,b,c) repeated here for convenience,

$$G_1(k) = P_1(k/k-1)H_1^T\left[H_1P_1(k/k-1)H_1^T + R\right]^{-1} , \quad (3.9a)$$

$$P_1(k/k) = \left[I - G_1(k)H_1\right]P_1(k/k-1) , \quad (3.9b)$$

and

$$P_1(k+1/k) = \Phi_1 P_1(k/k)\Phi_1^T + Q . \quad (3.9c)$$

Attention should be called to the fact that the presence of a deterministic forcing function in this problem changes slightly the form of Eq. (3.10), which can be rewritten as

$$\hat{\underline{x}}_1(k/k) = \Phi_1\hat{\underline{x}}_1(k-1/k-1) + \Gamma_1\underline{u}_1(k-1) + G_1(k)\Big[\underline{z}(k) - H_1\left[\Phi_1\hat{\underline{x}}_1(k-1/k-1) + \Gamma_1\underline{u}_1(k-1)\right]\Big]$$

to reflect the effect of $\underline{u}(k)$ on the prediction step.

The outputs of the computer simulation are shown in Tables III-2 through III-5, where numerical results versus time are tabulated for the gains, theoretical variance, experimental variance and experimental means of estimation error for both the complete and reduced filters. Figs. (3.3) through (3.6) show graphical representations comparing complete and reduced filters performance.

Table III-1

PHI MATRIX

0.9999	0.0499	0.0012	0.0000
-0.0066	0.9946	0.0480	0.0009
-0.3644	-0.2981	0.8871	0.0307
-12.2713	-10.1815	-3.9181	0.3043

GAMMA MATRIX

0.0001
0.0066
0.3644
12.2713

Q MATRIX

0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0008
0.0000	0.0000	0.0013	0.0447
0.0000	0.0008	0.0447	1.5059

VARIANCE OF FORCING NOISE = 0.01

VARIANCE OF MEASUREMENT NOISE = 0.25

INITIALIZATION - P(0/-1)

0.5000	0.0	0.0	0.0
0.0	2.0000	0.0	0.0
0.0	0.0	2.0000	0.0
0.0	0.0	0.0	4.0000

INITIALIZATION - X(0/-1)

0.0
0.0
0.0
0.0

T = 0.05 seconds.

Table III-2

GAINS FOR FULL AND REDUCED FILTERS

K	G(1,1)	G(2,1)	G(3,1)	G(4,1)	GR(1,1)	GR(2,1)
1	0.6667	0.0	0.0	0.0	0.6667	0.0
2	0.4071	0.2333	-0.2095	-7.2845	0.4071	0.2330
3	0.3097	0.4075	-0.6537	-9.4633	0.3102	0.4259
4	0.2677	0.5088	-1.1719	-9.0573	0.2708	0.5779
5	0.2477	0.5314	-1.6080	-7.0288	0.2569	0.6814
6	0.2349	0.4901	-1.8640	-4.4015	0.2540	0.7347
7	0.2227	0.4108	-1.9266	-1.9563	0.2542	0.7445
8	0.2087	0.3175	-1.8378	-0.0721	0.2537	0.7222
9	0.1928	0.2268	-1.6557	1.1851	0.2512	0.6798
10	0.1758	0.1474	-1.4305	1.9077	0.2464	0.6274
11	0.1584	0.0825	-1.1970	2.2348	0.2399	0.5720
12	0.1412	0.0322	-0.9760	2.2961	0.2321	0.5177
13	0.1247	-0.0051	-0.7780	2.1931	0.2237	0.4669
14	0.1093	-0.0315	-0.6072	1.9984	0.2151	0.4205
15	0.0950	-0.0490	-0.4637	1.7610	0.2064	0.3789
16	0.0819	-0.0596	-0.3457	1.5120	0.1979	0.3418
17	0.0701	-0.0647	-0.2505	1.2705	0.1897	0.3090
18	0.0596	-0.0660	-0.1752	1.0474	0.1819	0.2800
19	0.0502	-0.0643	-0.1167	0.8480	0.1745	0.2544
20	0.0420	-0.0606	-0.0723	0.6746	0.1675	0.2317
21	0.0348	-0.0556	-0.0395	0.5269	0.1609	0.2115
22	0.0286	-0.0499	-0.0161	0.4036	0.1547	0.1937
23	0.0233	-0.0438	-0.0001	0.3026	0.1488	0.1777
24	0.0188	-0.0378	0.0100	0.2215	0.1433	0.1635
25	0.0151	-0.0320	0.0157	0.1576	0.1382	0.1508
26	0.0120	-0.0266	0.0181	0.1084	0.1333	0.1393
27	0.0095	-0.0217	0.0181	0.0714	0.1287	0.1290
28	0.0075	-0.0174	0.0166	0.0444	0.1243	0.1197
29	0.0059	-0.0136	0.0141	0.0254	0.1203	0.1112
30	0.0047	-0.0105	0.0111	0.0127	0.1164	0.1036
31	0.0037	-0.0078	0.0080	0.0046	0.1127	0.0966
32	0.0031	-0.0057	0.0050	-0.0000	0.1092	0.0902
33	0.0026	-0.0041	0.0023	-0.0022	0.1059	0.0844
34	0.0022	-0.0028	-0.0001	-0.0028	0.1028	0.0790
35	0.0020	-0.0018	-0.0022	-0.0024	0.0998	0.0741
36	0.0018	-0.0011	-0.0038	-0.0015	0.0970	0.0696
37	0.0018	-0.0006	-0.0050	-0.0004	0.0943	0.0654
38	0.0017	-0.0003	-0.0060	0.0007	0.0917	0.0616
39	0.0017	-0.0001	-0.0066	0.0017	0.0893	0.0580
40	0.0017	0.0000	-0.0070	0.0024	0.0869	0.0547
41	0.0017	0.0001	-0.0073	0.0030	0.0847	0.0517
42	0.0017	0.0001	-0.0074	0.0033	0.0825	0.0488
43	0.0017	0.0001	-0.0074	0.0034	0.0804	0.0461
44	0.0017	0.0001	-0.0073	0.0033	0.0785	0.0437
45	0.0017	0.0000	-0.0072	0.0031	0.0765	0.0413
46	0.0017	0.0000	-0.0071	0.0028	0.0747	0.0392
47	0.0017	-0.0000	-0.0070	0.0025	0.0730	0.0371
48	0.0017	-0.0000	-0.0068	0.0022	0.0713	0.0352
49	0.0017	-0.0001	-0.0067	0.0018	0.0696	0.0334
50	0.0017	-0.0001	-0.0066	0.0015	0.0680	0.0318

G (4x1) represents the matrix of gains for the optimal filter.

GR(2x1) represents the matrix of gains for the reduced filter.

Table III-3

THEORETICAL VARIANCE OF ESTIMATION ERROR

K	PK(1,1)	PK(2,2)	PK(3,3)	PK(4,4)	PR(1,1)	PR(2,2)
1	0.1667	2.0000	2.0000	4.0000	0.1667	2.0000
2	0.1018	1.9603	1.7605	242.6279	0.1018	1.9558
3	0.0774	1.8002	2.3189	346.3362	0.0776	1.8684
4	0.0669	1.5182	3.9792	261.1479	0.0677	1.7325
5	0.0619	1.1771	5.7036	142.2309	0.0642	1.5559
6	0.0587	0.8478	6.6569	60.6997	0.0635	1.3561
7	0.0557	0.5745	6.6825	21.3209	0.0635	1.1534
8	0.0522	0.3709	6.0419	8.2223	0.0634	0.9640
9	0.0482	0.2307	5.0782	6.9855	0.0628	0.7970
10	0.0439	0.1400	4.0514	9.3412	0.0616	0.6557
11	0.0396	0.0847	3.1097	11.6978	0.0600	0.5391
12	0.0353	0.0529	2.3160	12.9672	0.0580	0.4442
13	0.0312	0.0361	1.6824	13.1242	0.0559	0.3676
14	0.0273	0.0282	1.1960	12.4778	0.0538	0.3058
15	0.0237	0.0255	0.8337	11.3650	0.0516	0.2559
16	0.0205	0.0254	0.5710	10.0516	0.0495	0.2155
17	0.0175	0.0264	0.3855	8.7171	0.0474	0.1827
18	0.0149	0.0275	0.2581	7.4696	0.0455	0.1557
19	0.0126	0.0283	0.1736	6.3652	0.0436	0.1336
20	0.0105	0.0286	0.1199	5.4260	0.0419	0.1152
21	0.0087	0.0282	0.0877	4.6526	0.0402	0.0999
22	0.0072	0.0273	0.0701	4.0336	0.0387	0.0870
23	0.0058	0.0259	0.0619	3.5512	0.0372	0.0762
24	0.0047	0.0241	0.0595	3.1854	0.0358	0.0670
25	0.0038	0.0221	0.0601	2.9159	0.0345	0.0591
26	0.0030	0.0199	0.0619	2.7237	0.0333	0.0524
27	0.0024	0.0177	0.0639	2.5918	0.0322	0.0466
28	0.0019	0.0155	0.0654	2.5057	0.0311	0.0416
29	0.0015	0.0133	0.0659	2.4529	0.0301	0.0373
30	0.0012	0.0114	0.0654	2.4236	0.0291	0.0335
31	0.0009	0.0096	0.0640	2.4099	0.0282	0.0302
32	0.0008	0.0081	0.0618	2.4059	0.0273	0.0272
33	0.0006	0.0067	0.0590	2.4071	0.0265	0.0247
34	0.0006	0.0056	0.0557	2.4105	0.0257	0.0224
35	0.0005	0.0046	0.0523	2.4141	0.0250	0.0204
36	0.0005	0.0039	0.0489	2.4167	0.0243	0.0186
37	0.0004	0.0033	0.0455	2.4177	0.0236	0.0170
38	0.0004	0.0028	0.0424	2.4170	0.0229	0.0155
39	0.0004	0.0024	0.0396	2.4146	0.0223	0.0142
40	0.0004	0.0022	0.0371	2.4109	0.0217	0.0131
41	0.0004	0.0020	0.0349	2.4062	0.0212	0.0120
42	0.0004	0.0018	0.0330	2.4008	0.0206	0.0111
43	0.0004	0.0017	0.0315	2.3951	0.0201	0.0102
44	0.0004	0.0017	0.0303	2.3894	0.0196	0.0094
45	0.0004	0.0017	0.0293	2.3839	0.0191	0.0087
46	0.0004	0.0016	0.0285	2.3787	0.0187	0.0081
47	0.0004	0.0016	0.0279	2.3741	0.0182	0.0075
48	0.0004	0.0016	0.0275	2.3700	0.0178	0.0070
49	0.0004	0.0016	0.0272	2.3664	0.0174	0.0065
50	0.0004	0.0016	0.0270	2.3635	0.0170	0.0060

PK represents the diagonal elements of the theoretical covariance matrix for the optimal filter.

PR represents the diagonal elements of the theoretical covariance matrix for the reduced filter.

Table III-4

EXPERIMENTAL VARIANCE OF ESTIMATION ERROR

K	VK(1,1)	VK(2,2)	VK(3,3)	VK(4,4)	VR(1,1)	VR(2,2)
1	0.1510	1.9066	1.9733	4.0880	0.1486	1.9041
2	0.1022	1.8632	1.7310	225.4341	0.1008	1.8617
3	0.0807	1.7518	2.2373	326.4941	0.0794	1.7496
4	0.0667	1.5112	3.8145	257.3462	0.0652	1.5140
5	0.0677	1.1743	5.6299	138.0874	0.0663	1.2011
6	0.0529	0.8372	6.0810	60.2167	0.0511	0.9078
7	0.0518	0.5928	6.5273	23.2761	0.0504	0.7632
8	0.0487	0.3939	6.0208	8.7271	0.0492	0.7093
9	0.0464	0.2513	5.1600	7.2510	0.0497	0.7732
10	0.0457	0.1612	4.4359	9.6855	0.0542	0.9210
11	0.0420	0.1002	3.5490	12.4134	0.0570	1.0924
12	0.0391	0.0617	2.7683	14.0120	0.0559	1.1048
13	0.0368	0.0401	2.0579	14.2776	0.0615	1.1834
14	0.0297	0.0300	1.3831	14.0637	0.0592	1.2549
15	0.0264	0.0256	0.9971	11.9529	0.0587	1.2175
16	0.0226	0.0247	0.6811	12.3539	0.0593	1.1740
17	0.0188	0.0260	0.4289	9.7735	0.0585	1.0867
18	0.0153	0.0261	0.2742	7.4905	0.0616	1.0225
19	0.0131	0.0271	0.1830	6.0835	0.0639	0.9195
20	0.0112	0.0269	0.1292	5.5882	0.0661	0.8078
21	0.0094	0.0274	0.0892	5.2477	0.0694	0.7044
22	0.0074	0.0252	0.0707	4.1038	0.0661	0.5994
23	0.0061	0.0240	0.0651	3.5547	0.0661	0.4966
24	0.0047	0.0216	0.0634	2.8311	0.0649	0.4195
25	0.0039	0.0202	0.0596	3.2233	0.0640	0.3411
26	0.0032	0.0185	0.0569	2.5204	0.0593	0.2732
27	0.0025	0.0166	0.0547	2.7723	0.0555	0.2257
28	0.0020	0.0149	0.0546	2.8033	0.0566	0.1949
29	0.0016	0.0130	0.0550	2.6636	0.0542	0.1729
30	0.0013	0.0114	0.0581	2.0651	0.0510	0.1560
31	0.0010	0.0098	0.0584	2.4213	0.0474	0.1489
32	0.0008	0.0083	0.0567	2.7746	0.0459	0.1498
33	0.0007	0.0071	0.0539	2.5252	0.0435	0.1523
34	0.0006	0.0060	0.0490	2.1855	0.0391	0.1563
35	0.0005	0.0050	0.0435	2.5705	0.0378	0.1674
36	0.0005	0.0042	0.0420	2.2536	0.0363	0.1752
37	0.0004	0.0035	0.0437	2.5342	0.0349	0.1817
38	0.0004	0.0029	0.0446	2.3070	0.0350	0.1891
39	0.0004	0.0025	0.0421	2.0512	0.0378	0.1947
40	0.0004	0.0022	0.0366	2.4148	0.0393	0.1985
41	0.0004	0.0019	0.0300	2.2901	0.0403	0.2024
42	0.0004	0.0018	0.0256	2.5146	0.0405	0.2035
43	0.0004	0.0017	0.0241	2.2052	0.0411	0.2022
44	0.0004	0.0016	0.0261	2.3978	0.0429	0.2016
45	0.0004	0.0015	0.0300	1.9640	0.0439	0.1990
46	0.0004	0.0014	0.0317	1.9171	0.0476	0.1957
47	0.0004	0.0014	0.0307	2.2013	0.0494	0.1914
48	0.0004	0.0014	0.0299	2.3214	0.0500	0.1835
49	0.0004	0.0015	0.0281	2.2047	0.0532	0.1772
50	0.0004	0.0016	0.0229	2.8952	0.0570	0.1698

VK represents the diagonal elements of the experimental covariance matrix for the optimal process.

VR represents the diagonal elements of the experimental covariance matrix for the reduced process.

Table III-5

EXPERIMENTAL MEANS OF ESTIMATION ERROR

K	MEANE1	MEANE2	MEANE3	MEANE4	MEANE1R	MEANE2R
1	0.0496	0.0500	-0.0350	-0.1543	0.0496	0.0500
2	0.0372	0.0392	-0.0572	-0.6276	0.0372	0.0392
3	0.0356	0.0307	-0.0903	-0.7992	0.0358	0.0429
4	0.0380	0.0271	-0.1340	-0.7676	0.0391	0.0662
5	0.0395	0.0203	-0.1637	-0.3713	0.0425	0.0973
6	0.0333	-0.0030	-0.1219	-0.0534	0.0387	0.1099
7	0.0299	-0.0146	-0.0909	0.2345	0.0387	0.1346
8	0.0044	-0.0565	0.1352	0.1478	0.0122	0.0810
9	-0.0017	-0.0536	0.1711	0.0767	0.0089	0.1003
10	0.0049	-0.0373	0.1013	0.1589	0.0229	0.1579
11	0.0142	-0.0265	0.0168	-0.0124	0.0414	0.2121
12	0.0021	-0.0281	0.0951	-0.0444	0.0257	0.1751
13	-0.0028	-0.0232	0.1136	-0.1564	0.0209	0.1623
14	-0.0072	-0.0168	0.1219	-0.3115	0.0155	0.1447
15	-0.0086	-0.0108	0.1078	-0.4010	0.0148	0.1327
16	-0.0058	-0.0083	0.0729	-0.3752	0.0232	0.1322
17	-0.0028	-0.0080	0.0511	0.0015	0.0320	0.1267
18	0.0020	-0.0112	0.0313	-0.0783	0.0461	0.1255
19	0.0055	-0.0150	0.0153	-0.0996	0.0572	0.1156
20	0.0044	-0.0137	0.0109	-0.1052	0.0511	0.0790
21	0.0040	-0.0136	0.0142	0.2029	0.0476	0.0468
22	0.0014	-0.0094	0.0236	0.1062	0.0321	0.0012
23	0.0022	-0.0103	0.0273	0.0610	0.0343	-0.0198
24	0.0017	-0.0089	0.0262	-0.0824	0.0284	-0.0492
25	0.0011	-0.0072	0.0235	-0.0285	0.0199	-0.0795
26	0.0008	-0.0063	0.0215	-0.0504	0.0143	-0.1043
27	0.0007	-0.0056	0.0207	-0.0033	0.0100	-0.1258
28	0.0006	-0.0048	0.0263	0.1783	0.0050	-0.1455
29	0.0001	-0.0027	0.0329	0.1111	-0.0078	-0.1700
30	0.0002	-0.0014	0.0292	-0.2192	-0.0114	-0.1832
31	0.0003	-0.0006	0.0215	-0.1151	-0.0128	-0.1929
32	0.0004	0.0002	0.0170	-0.0758	-0.0172	-0.2037
33	0.0005	0.0008	0.0080	-0.2516	-0.0234	-0.2143
34	0.0004	0.0011	0.0012	-0.0467	-0.0344	-0.2271
35	0.0004	0.0012	0.0022	0.0639	-0.0455	-0.2379
36	0.0005	0.0014	0.0016	-0.0675	-0.0514	-0.2430
37	0.0006	0.0014	-0.0011	-0.0404	-0.0574	-0.2467
38	0.0006	0.0012	-0.0069	-0.1685	-0.0654	-0.2506
39	0.0006	0.0007	-0.0093	0.0404	-0.0711	-0.2520
40	0.0006	0.0003	-0.0057	0.0877	-0.0785	-0.2533
41	0.0007	0.0002	-0.0006	0.1207	-0.0795	-0.2494
42	0.0007	0.0003	0.0027	0.0151	-0.0876	-0.2489
43	0.0008	0.0004	0.0038	0.0374	-0.0882	-0.2432
44	0.0008	0.0006	0.0026	-0.0628	-0.0905	-0.2381
45	0.0008	0.0007	0.0004	-0.0407	-0.0978	-0.2354
46	0.0008	0.0007	0.0035	0.1341	-0.1020	-0.2307
47	0.0007	0.0010	0.0075	0.0162	-0.1105	-0.2277
48	0.0008	0.0014	0.0069	-0.0321	-0.1128	-0.2212
49	0.0009	0.0017	0.0066	0.0174	-0.1143	-0.2143
50	0.0010	0.0020	0.0053	-0.0515	-0.1149	-0.2071

MEANEI through MEANE4 represent the mean of estimation error in the complete state vector by means of optimal filtering.

MEANE1R and MEANE2R represent the mean of estimation error in states x_1 and x_2 by means of a reduced filter.

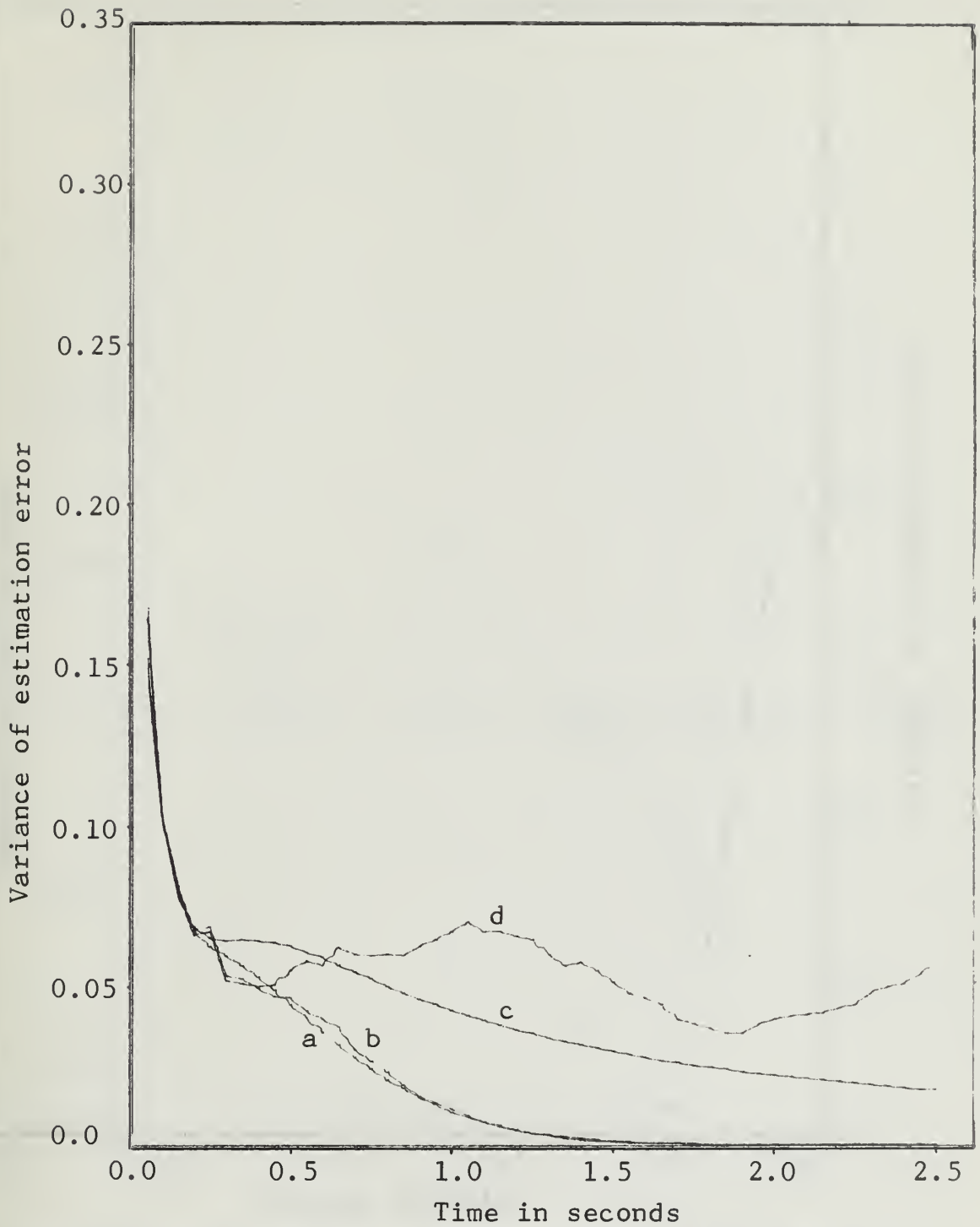


Fig. 3.3 - Graphical plot of the variance of estimation error in x_1 vs. time. Curves a and b show the variance of estimation error, both theoretical and experimental, for the optimal estimation process. Curves c and d show the corresponding theoretical and experimental variances for the reduced case.

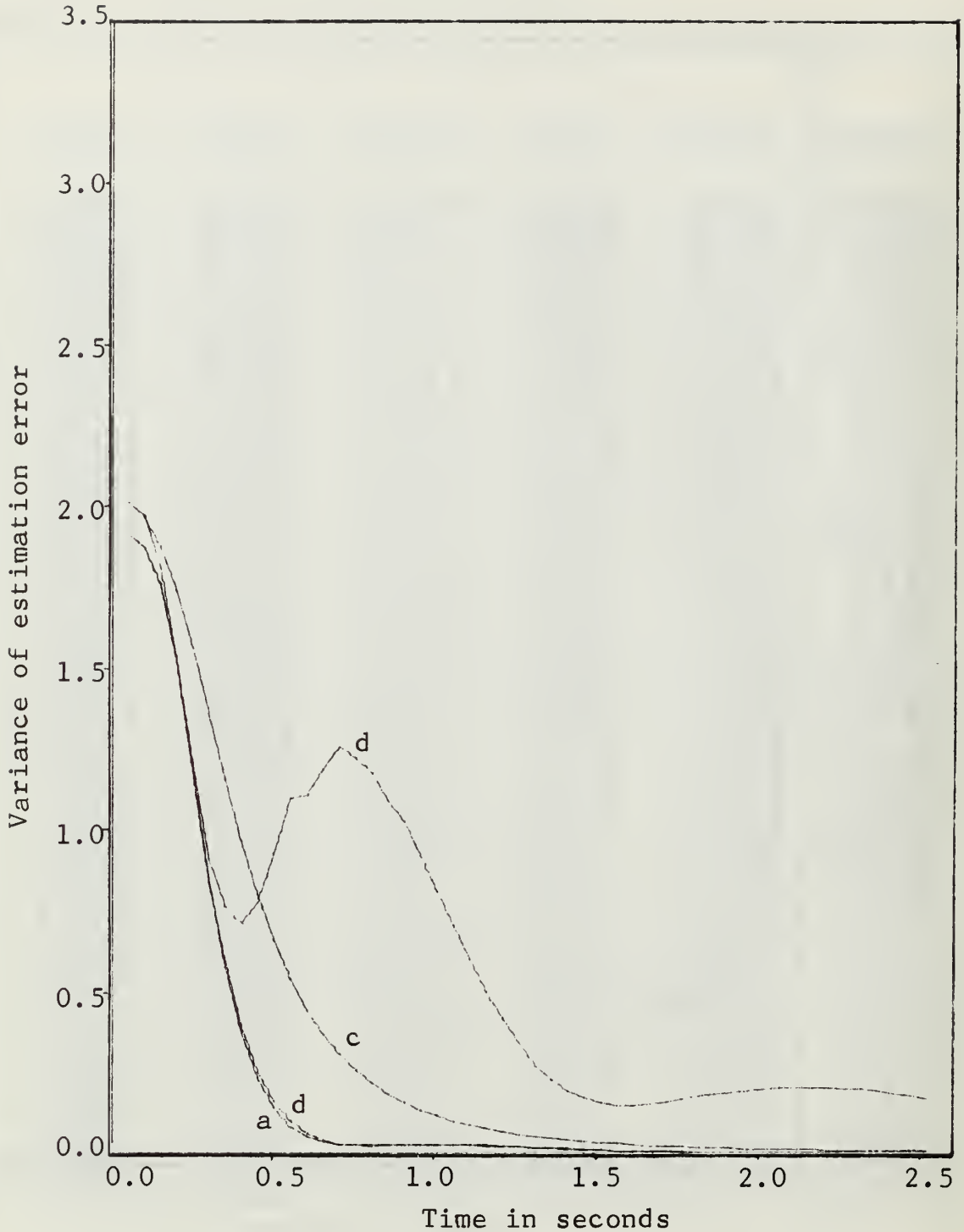


Fig. 3.4 - Graphical plot of the variance of est. error in x_2 vs. time. Curves a and b show the variance of estimation error, both theoretical and experimental, for the optimal estimation process. Curves c and d show the corresponding theoretical and experimental variances for the reduced case.

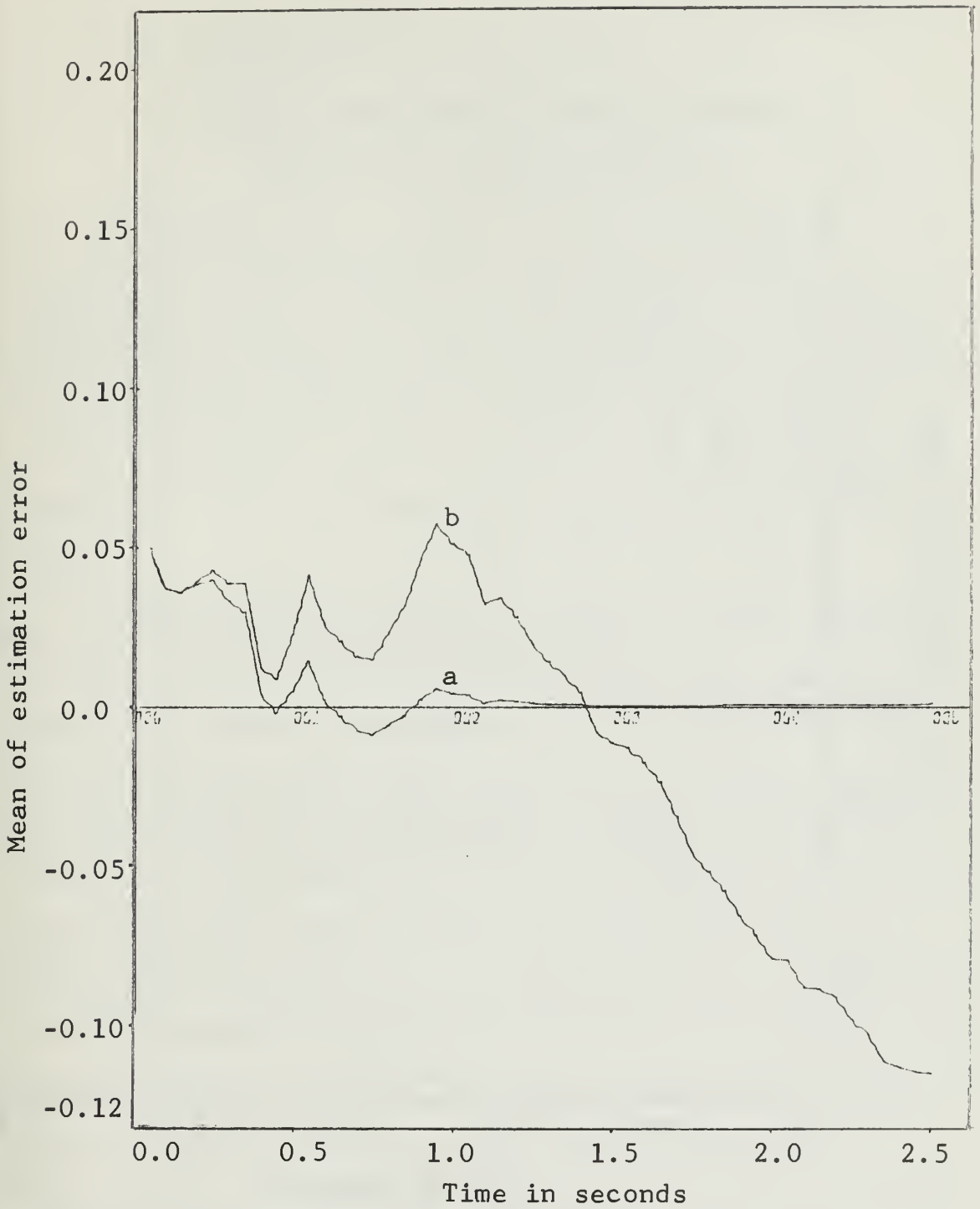


Fig. 3.5 - Graphical plot of the means of estimation error in x_1 vs. time. Curve a shows the mean of estimation error for the optimal Kalman filter, and curve b shows that for the reduced filter.

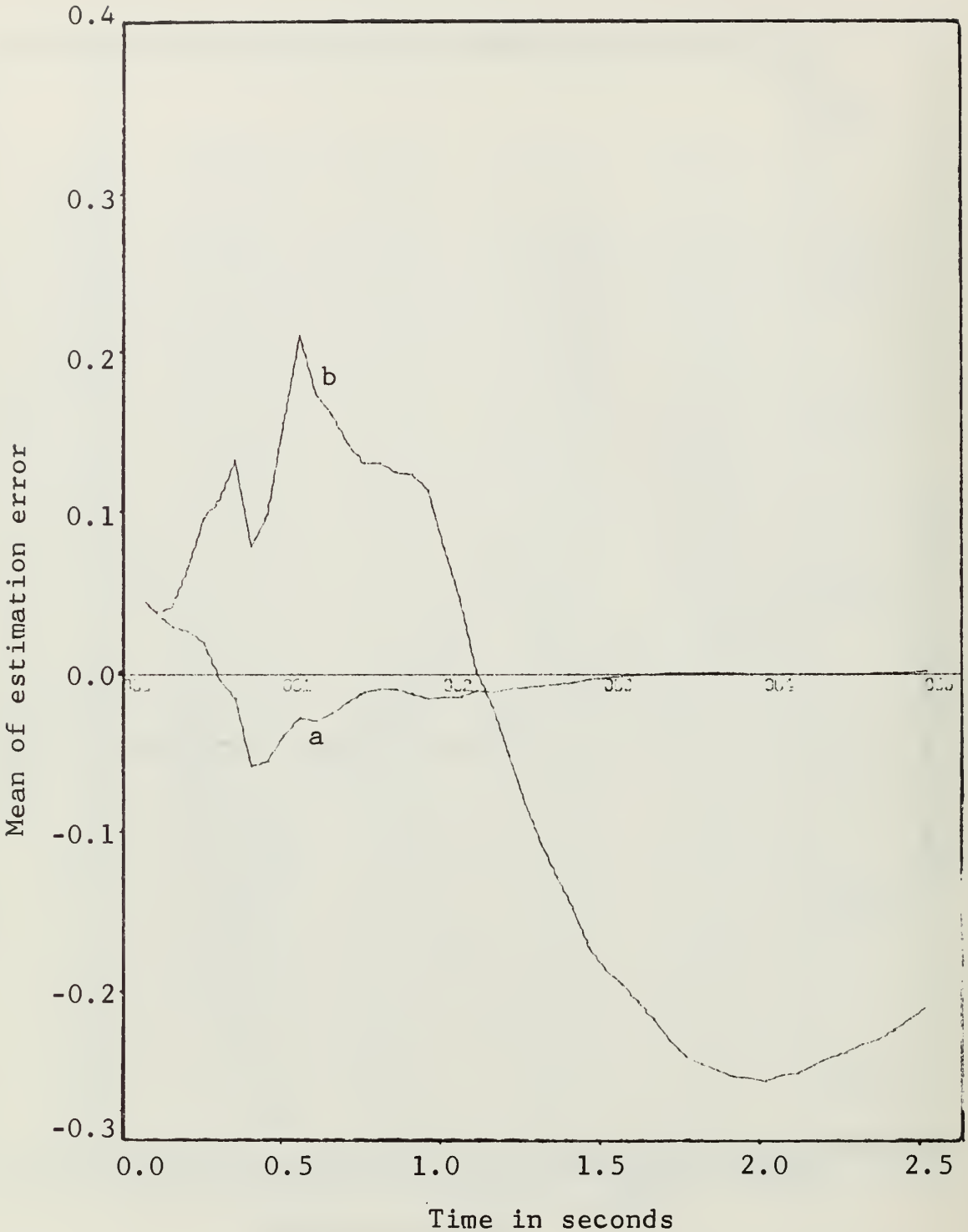


Fig. 3.6 - Graphical plot of the means of estimation error in x_2 vs. time. Curve a shows the mean of estimation error for the optimal Kalman filter, and curve b shows that for the reduced filter.

IV. IMPLEMENTATION OF SUB-OPTIMAL LINEAR FILTERS FOR DISCRETE DYNAMIC PROCESSES USING MATRIX PSEUDO-INVERSION

In the previous chapter a complete discussion of the implementation of a discrete, reduced model Kalman filter was presented, where the idea of partitioning the dynamics of the original plant was principal. In this chapter, a different approach to the same problem is presented, based upon the work of Meditch (Ref.3) for the continuous case.

Applying the same idea of reduction of the original process, it is partitioned for this case into two or more reduced $\underline{y}_i$ vector components (i = 1,2 ... M), with each new vector containing the same number of states. This new partitioning is accomplished by using matrix pseudo-inversion (see Appendix A). Once the original system has been broken down into M [1] reduced-order sub-optimal filters, then the sub-optimal estimates for the original process are obtained by recombination of the outputs from all the sub-optimal filters. Section A presents the derivation of the discrete sub-optimal filter formulation. These results are extended to Section B, where expressions

---

[1] The order (n) of the original system is equal to M times the order (r) of the reduced plants, that is $n = M \cdot r$.

for the sub-optimal estimates, theoretical sub-optimal estimation error and theoretical sub-optimal covariance of estimation error are derived. Finally in Section C an example is solved to illustrate the technique.

A. DERIVATION OF THE SUB-OPTIMAL FILTER EQUATIONS

As in Chapter III-A it is assumed that the system dynamics are represented by Eq. (2.9), without considering the presence of the deterministic part of the forcing vector, that is

$$\underline{x}(k+1) = \Phi\underline{x}(k) + \Gamma\underline{w}(k) \ ,$$

with measurement

$$\underline{z}(k) = H\underline{x}(k) + \underline{v}(k) \ ,$$

where as before, $\underline{w}(k)$ and $\underline{v}(k)$ are random gaussian functions fully described by Eqs. (2.11) through (2.16).

If, as discussed in Chapter I, it is desired to estimate only r states, where r<n, then the original vector $\underline{x}(k)$ can be partitioned or transformed into two or more vectors by using a set of constant matrices

$$A_i, \quad i = 1,2 \ldots M$$

with A_i having dimensions (r x n). Notice that each A_i must have r rows and n columns. Then it is possible to define a new set of reduced vectors

$$\underline{y}_i(k) = A_i \underline{x}(k), \quad i = 1,2 \ldots M \tag{4.1}$$

with $\underline{y}_i(k)$ representing a (r x 1) vector.

Now the concept of the Generalized Inverse[1] of a matrix (pseudo-inversion) is applied to the implementation of the sub-optimal filters. A matrix A^+ [2] is considered to be the pseudo-inverse of a rectangular matrix A if

$$AA^+A = A \ , \tag{4.2}$$

where in general A is a (s x t) matrix and A^+ is a (t x s) matrix.

The selection of the set of constant matrices A_i will have as a necessary requirement for the solution of the filter implementation and the proper recovery of the sub-optimal estimates from the combination of the filter outputs, that

$$\sum_{i=1}^{M} A_i^+ A_i = I \ , \tag{4.3}$$

---

1 See Deutsch (Ref.10) and Penrose (Ref.8) for a complete discussion about matrix pseudo-inversion.

2 A^+ stands for the pseudo-inverse of A.

where I is the identity matrix. This requirement will be clarified after the derivation which follows.

If Eq. (4.1) is premultiplied by A_i^+, $i = 1,2 \ldots M$ and summation over all the i's is performed, then

$$\sum_{i=1}^{M} A_i^+ \underline{y}_i(k) = \sum_{i=1}^{M} A_i^+ A_i \underline{x}(k) ,$$

from which it is clearly seen, applying the requirement established by Eq. (4.3), that

$$\underline{x}(k) = \sum_{i=1}^{M} A_i^+ \underline{y}_i(k) . \tag{4.4}$$

This last equation provides the key for recovery of the original system state vector.

Now if the original discrete system equation is premultiplied by A_i, $i = 1,2 \ldots M$,

$$A_i \underline{x}(k+1) = A_i \Phi \underline{x}(k) + A_i \Gamma \underline{w}(k) , \tag{4.5}$$

and if Eqs. (4.1) and (4.4) are substituted in Eq. (4.5), the latter becomes

$$\underline{y}_i(k+1) = A_i \Phi \sum_{j=1}^{M} A_j^+ \underline{y}_j(k) + A_i \Gamma \underline{w}(k), \quad i = 1,2 \ldots M \tag{4.6}$$

Expression (4.6) represents a set of M reduced, discrete system equations, each of order r.

The measurement process can be written assuming that all the measurements that were available for the complete filter are available too for the reduced filters.

$$\underline{z}(k) = H\underline{x}(k) + \underline{v}(k) \ ,$$

becomes after replacement of $\underline{x}(k)$ by Eq. (4.4),

$$\underline{z}(k) = H \sum_{j=1}^{M} A_j^{+}\underline{y}_j(k) + \underline{v}(k) \ . \qquad (.47)$$

Fig. (4.1) will help to better visualize the partitioned system.

In order to facilitate the following derivations, the optimal Kalman filter estimation Eq. (3.8) may be rewritten in a slightly different form, that is

$$\hat{\underline{x}}(k/k) = \Phi\hat{\underline{x}}(k\text{-}1/k\text{-}1) + G(k)\left[\underline{z}(k) - \hat{\underline{z}}(k/k\text{-}1)\right] \ , \qquad (4.8)$$

where $\hat{\underline{z}}(k/k\text{-}1)$ represents the expected value of $\underline{z}(k)$ given all measurements up to time (k-1), or

$$\hat{\underline{z}}(k/k\text{-}1) = E\left[\underline{z}(k)\right] \quad \text{given} \quad \underline{z}(k\text{-}1) \ .$$

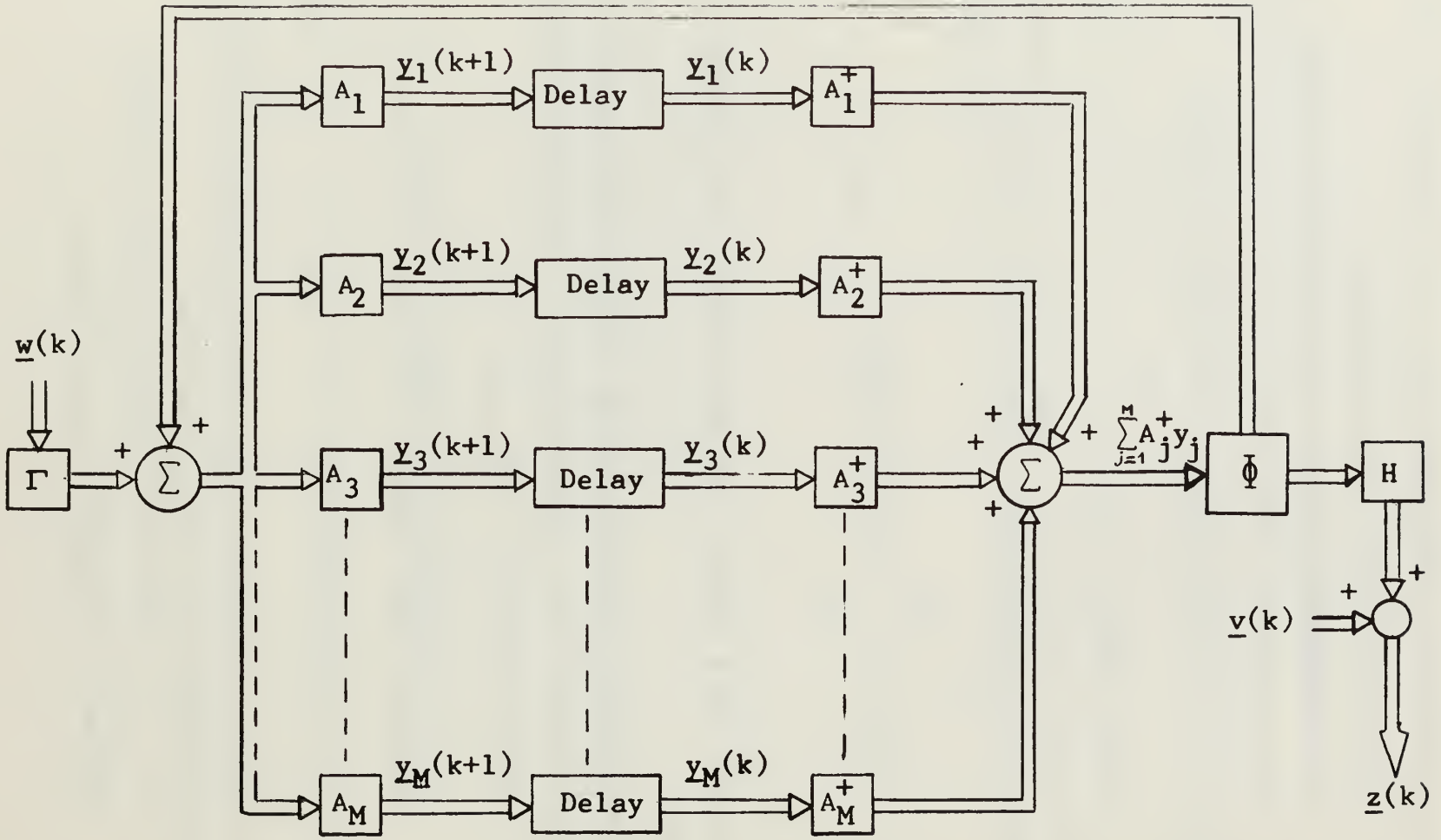


Fig. 4.1 - Block diagram representation of the original system partitioned into M sub-systems $\underline{y}_i$ by means of matrix pseudo-inversion.

From Eq. (4.7), however,

$$\underline{z}(k) = H \sum_{j=1}^{M} A_j^+ \underline{y}_j(k) + \underline{v}(k) \ . \tag{4.7}$$

Therefore,

$$\hat{\underline{z}}(k/k-1) = H \sum_{j=1}^{M} A_j^+ E\left[\underline{y}_j(k)\right] \quad \text{given } \underline{z}(k-1)$$

or

$$\hat{\underline{z}}(k/k-1) = H \sum_{j=1}^{M} A_j^+ \hat{\underline{y}}_j(k/k-1) \ . \tag{4.9}$$

Considering further that

$$\hat{\underline{x}}(k/k-1) = \Phi\hat{\underline{x}}(k-1/k-1) \ , \tag{4.10}$$

where $\hat{\underline{x}}(k/k-1)$ represents the expected value of $\underline{x}(k)$ given all measurements up to time (k-1), and premultiplying Eq. (4.10) by A_i, 1,2 ... M,

$$A_i\hat{\underline{x}}(k/k-1) = A_i\Phi\hat{\underline{x}}(k-1/k-1) \ . \tag{4.11}$$

By extension of Eq. (4.1)

$$A_i\hat{\underline{x}}(k/k-1) = \hat{\underline{y}}_i(k/k-1) \ ,$$

and similarly by proper use of Eq. (4.4)

$$\hat{\underline{x}}(k\text{-}1/k\text{-}1) = \sum_{j=1}^{M} A_j^+ \hat{\underline{y}}_j(k\text{-}1/k\text{-}1) ,$$

so Eq. (4.11) becomes

$$\hat{\underline{y}}_i(k/k\text{-}1) = A_i \bar{\Phi} \sum_{j=1}^{M} A_j^+ \hat{\underline{y}}_j(k\text{-}1/k\text{-}1) . \qquad (4.12)$$

Basing the implementation of the reduced filters upon the dynamics of the set of systems described by Eq. (4.6), a set of M reduced-filter estimation equations may be stated by induction from Eq. (4.8), as

$$\hat{\underline{y}}_i(k/k) = \hat{\underline{y}}_i(k/k\text{-}1) + G_i(k)\left[\underline{z}(k) - \hat{\underline{z}}(k/k\text{-}1)\right] .$$

Replacing for $\hat{\underline{y}}_i(k/k\text{-}1)$ and $\hat{\underline{z}}(k/k\text{-}1)$ Eqs. (4.12) and (4.9), gives

$$\hat{\underline{y}}_i(k/k) = A_i \bar{\Phi} \sum_{j=1}^{M} A_i^+ \hat{\underline{y}}_j(k\text{-}1/k\text{-}1) + G_1(k)\left[\underline{z}(k) - H \sum_{j=1}^{M} A_j^+ \hat{\underline{y}}_j(k/k\text{-}1)\right] \qquad (4.13)$$

with i = 1,2 ... M.

This expression represents a set of M r^{th}-order coupled reduced filters, whose outputs $\hat{\underline{y}}_i(k/k)$ will be recombined

in a proper way in order to get the sub-optimal estimates $\hat{\underline{x}}(k/k)$, where $\hat{\underline{x}}(k/k)$ is an (n x 1) vector (see Section IV-B).

Referring to the optimal gain and theoretical variance expressions for the Kalman filter (Eqs. 3.7a,b,c), the equivalent M subsets of sub-optimal equations for each one of the reduced filters may be expressed, by definition, as

$$G_i(k) = P_i(k/k-1)H_i^T\left[H_iP_i(k/k-1)H_i^T+R\right]^{-1} , \qquad (4.14a)$$

where

$$H_i = H\,A_i^+ \quad \text{(see Eq. 4.7)}$$

corresponds to the i^{th} subsystem contribution to the measurement vector;

$$P_i(k/k) = \left[I - G_i(k)H_i\right]P_i(k/k-1) , \qquad (4.14b)$$

and

$$P_i(k+1/k) = \Phi_iP_i(k/k)\Phi_i^T + Q_i \qquad (4.14c)$$

where

$$\Phi_i = A_i\Phi A_i^+ \quad \text{(see Eq. 4.6)}$$

$$Q_i = A_i\Gamma E\left[\underline{w}(k)\underline{w}^T(k)\right]\Gamma^TA_i^T$$

$$= A_iQA_i^T$$

and

$$\Gamma_i = A_i \Gamma \ .$$

It is important to notice that although the M estimation equations and the measurement equation are coupled for all the M reduced filters, the gain and covariance equations for the i^{th} filter are truly uncoupled from the other j = 1,2 ... M filters, where $j \neq i$.

Eqs. (4.14) complete the implementation of the M sub-optimal filters. The filtering process will be visualized better by means of Fig. (4.2).

The initialization values for the sub-optimal filters are derived from the statistics used to initialize the complete optimal filter in the following way. Given the values for P(0/-1) and knowing that

$$P_i(0/-1) = E\left[\underline{y}_i(0)\underline{y}_i^T(o)\right]$$

and

$$\underline{y}_i(0) = A_i\underline{x}(0) \ ,$$

then

$$\begin{aligned} P_i(0/-1) &= E\left[A_i\underline{x}(0)\underline{x}^T(0)A_i^T\right] \\ &= A_i E\left[\underline{x}(0)\underline{x}^T(0)\right]A_i^T \\ &= A_i P(0/-1)A_i^T \ . \end{aligned} \qquad (4.15)$$

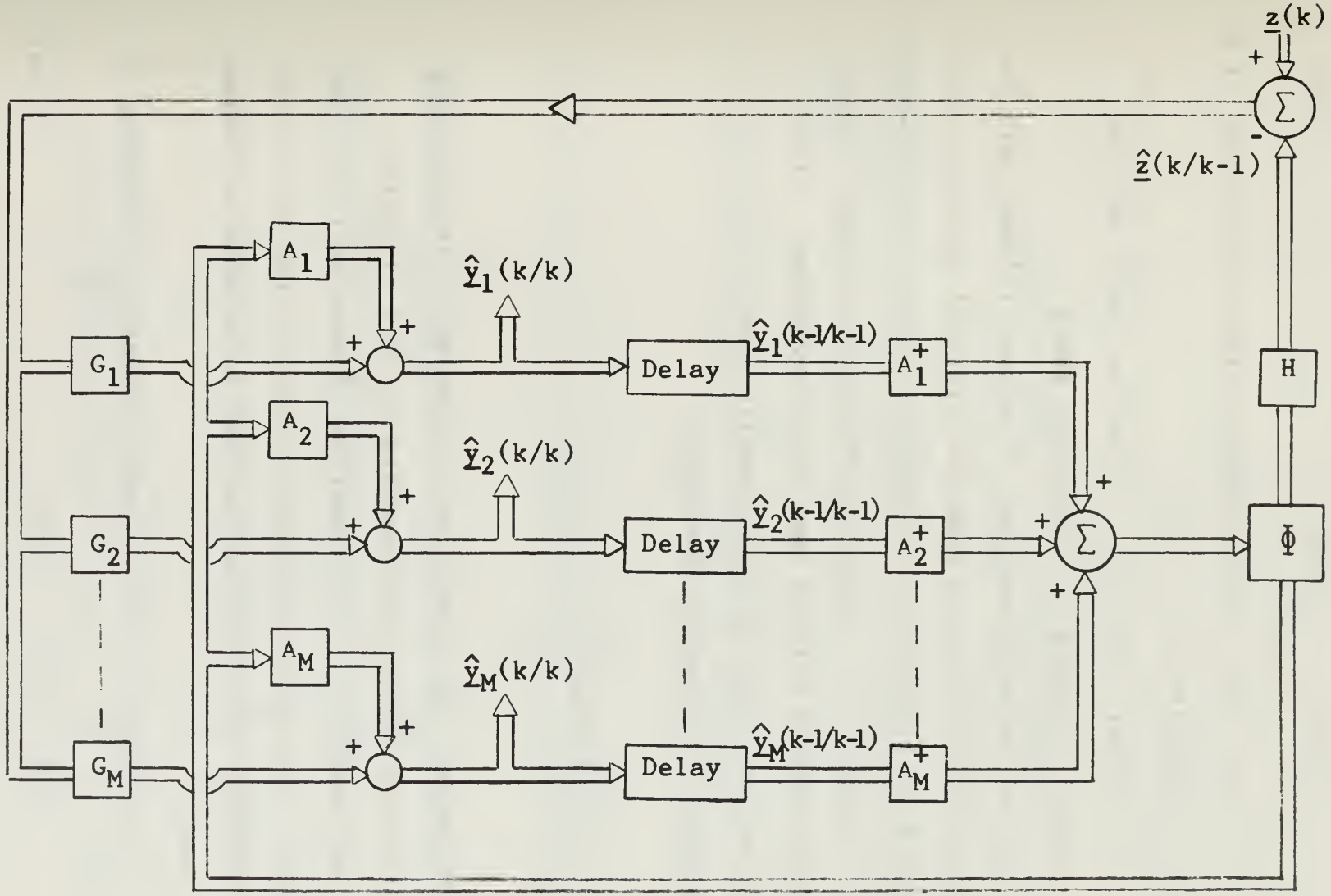


Fig. 4.2 - Block diagram representation of the M sub-optimal filters, showing the outputs from each reduced filter, $\hat{y}_i(k/k)$.

By the same reasoning,

$$\hat{\underline{y}}_i(0/-1) = A_i \hat{\underline{x}}(0/-1) \tag{4.16}$$

will be used as the initialization values for the set of M reduced estimation Eqs. (4.13).

B. SUB-OPTIMAL ESTIMATES, ESTIMATION ERROR AND SUB-OPTIMAL VARIANCE OF ESTIMATION ERROR

In the preceding section, M reduced filters were implemented after breaking down the original process into M reduced (r x 1) vectors. In order to obtain the sub-optimal estimates of the original process, the outputs $\hat{\underline{y}}_i(k/k)$ from the reduced filters have to be recombined. This is easily accomplished by proper application of Eq. (4.4) to the filter's estimates, whereupon

$$\hat{\hat{\underline{x}}}(k/k) = \sum_{j=1}^{M} A_j^+ \hat{\underline{y}}_j(k/k) \tag{4.17}$$

will represent the sub-optimal estimates of the original process, with $\hat{\hat{\underline{x}}}(k/k)$ representing an (n x 1) vector.

The remainder of this section is devoted to the development of a matrix difference equation for the iterative computation of the sub-optimal covariance of estimation error.

If the error of sub-optimal estimation is defined as

$$\bar{\underline{e}}(k) = \underline{x}(k) - \hat{\hat{\underline{x}}}(k/k) \tag{4.18a}$$

then using Eq. (4.17) it becomes

$$\bar{\underline{e}}(k) = \underline{x}(k) - \sum_{j=1}^{M} A_j^+ \hat{\underline{y}}_j(k/k) \ , \qquad (4.18b)$$

or

$$\bar{\underline{e}}(k+1) = \underline{x}(k+1) - \sum_{j=1}^{M} A_j^+ \hat{\underline{y}}_j(k+1/k+1) \ . \qquad (4.18c)$$

Substitution of Eqs. (4.13) and (4.7) into Eq. (4.18c) gives

$$\bar{\underline{e}}(k+1) = \underline{x}(k+1) - \sum_{j=1}^{M} A_j^+ \Big[A_i \bar{\Phi} \sum_{j=1}^{M} A_j^+ \hat{\underline{y}}_j(k/k)$$

$$+G_i(k+1)\Big[H \sum_{j=1}^{M} A_j^+ \underline{y}_j(k+1)+\underline{v}(k+1)-H\bar{\Phi} \sum_{j=1}^{M} A_j^+\hat{\underline{y}}_j(k/k)\Big] \Big] \ .$$

Replacing $\sum_{j=1}^{M} A_j^+\underline{y}_j(k+1)$ with $\underline{x}(k+1)$, and substituting $\underline{x}(k+1) = \bar{\Phi}\underline{x}(k) + \Gamma\underline{w}(k)$ from Eq. (3.1), then

$$\bar{\underline{e}}(k+1) = \bar{\Phi}\underline{x}(k) + \Gamma\underline{w}(k) - \sum_{j=1}^{M} A_j^+\Big[A_i\bar{\Phi} \sum_{j=1}^{M} A_j^+\hat{\underline{y}}_j(k/k)$$

$$+G_i(k+1)\Big[H\bar{\Phi}\underline{x}(k) + H\Gamma\underline{w}(k) + v(k+1) - H\bar{\Phi} \sum_{j=1}^{M} A_j^+\hat{\underline{y}}_j(k/k)\Big]\Big] \ .$$

Expansion of this expression for the sub-optimal error leads to

$$\underline{\bar{e}}(k+1) = \Phi\underline{x}(k) + \Gamma\underline{w}(k) - \sum_{j=1}^{M} A_j^+ A_j \Phi \sum_{j=1}^{M} A_j^+ \underline{\hat{y}}_j(k/k)$$

$$- \sum_{j=1}^{M} A_j^+ G_j(k+1) H \Phi \underline{x}(k) - \sum_{j=1}^{M} A_j^+ G_j(k+1) H \Gamma \underline{w}(k)$$

$$- \sum_{j=1}^{M} A_j^+ G_j(k+1) \underline{v}(k+1) + \sum_{j=1}^{K} A_j^+ G_j(k+1) H \Phi \sum_{j=1}^{M} A_j^+ \underline{\hat{y}}_j(k/k).$$

Collecting terms,

$$\underline{\bar{e}}(k+1) = \left[\Phi - \sum_{j=1}^{M} A_j^+ G_j(k+1) H \Phi\right]\left[\underline{x}(k) - \sum_{j=1}^{M} A_j^+ \underline{\hat{y}}_j(k/k)\right]$$

$$+ \left[I - \sum_{j=1}^{M} A_j^+ G_j(k+1) H\right] \Gamma \underline{w}(k) - \sum_{j=1}^{M} A_j^+ G_j(k+1) \underline{v}(k+1).$$

Substitution of Eq. (4.18b) for $\underline{\bar{e}}(k)$ gives

$$\underline{\bar{e}}(k+1) = \left[I - \sum_{j=1}^{M} A_j^+ G_j(k+1) H\right] \Phi \underline{\bar{e}}(k) + \left[I - \sum_{j=1}^{M} A_j^+ G_j(k+1) H\right] \Gamma \underline{w}(k)$$

$$- \sum_{j=1}^{M} A_j^+ G_j(k+1) \underline{v}(k+1). \qquad (4.19)$$

Eq. (4.19) represents a theoretical form for the error of sub-optimal estimation which can be useful in establishing the steady-state estimation error for the sub-optimal filter.

If the sub-optimal theoretical variance of estimation error is defined as

$$\bar{S}(k) = E\left[\bar{\underline{e}}(k)\bar{\underline{e}}^T(k)\right] \tag{4.20}$$

where $\bar{S}(k)$ is an (n x n)matrix, then substitution of Eq. (4.19) for $\bar{\underline{e}}(k)$ gives

$$\bar{S}(k+1) = E\Bigg\{\Bigg[\Big[I - \sum_{j=1}^{M} A_j^+ G_j(k+1)H\Big]\Phi\bar{\underline{e}}(k) + \Big[I - \sum_{j=1}^{M} A_j^+ G_j(k+1)H\Big]\Gamma\underline{w}(k)$$

$$- \sum_{j=1}^{M} A_j^+ G_j(k+1)\underline{v}(k+1)\Bigg]\Big[\text{ same terms }\Big]^T\Bigg\} \quad .$$

Carrying out the indicated matrix operations term by term yields

$$\bar{S}(k+1) = E\Bigg\{\Big[I - \sum_{j=1}^{M} A_j^+ G_j(k+1)H\Big]\Phi\bar{\underline{e}}(k)\bar{\underline{e}}^T(k)\Phi^T\Big[I - \sum_{j=1}^{M} A_j^+ G_j(k+1)H\Big]^T$$

$$+ \Big[I - \sum_{j=1}^{M} A_j^+ G_j(k+1)H\Big]\Gamma\underline{w}(k)\underline{w}^T(k)\Gamma^T\Big[I - \sum_{j=1}^{M} A_j^+ G_j(k+1)H\Big]^T$$

$$+ \sum_{j=1}^{M} A_j^+ G_j(k+1)\underline{v}(k+1)\underline{v}^T(k+1)\sum_{j=1}^{M} G_j^T(k+1)A_j^{+T} + \text{cross products}\Bigg)$$

Given the statistics of the problem, all the cross products will yield zero expectation. $\bar{S}(k+1)$ then becomes

$$\bar{S}(k+1) = \left[I - \sum_{j=1}^{M} A_j^+ G_j(k+1)H\right] \bar{\Phi} E\left[\underline{\bar{e}}(k)\underline{\bar{e}}^T(k)\right] \bar{\Phi}^T \left[I - \sum_{j=1}^{M} A_j^+ G_j(k+1)H\right]^T$$

$$+\left[I - \sum_{j=1}^{M} A_j^+ G_j(k+1)H\right] \Gamma E\left[\underline{w}(k)\,\underline{w}^T(k)\right] \Gamma^T \left[I - \sum_{j=1}^{M} A_j^+ G_j(k+1)H\right]^T$$

$$+\sum_{j=1}^{M} A_j^+ G_j(k+1) E\left[\underline{v}(k+1)\underline{v}^T(k+1)\right] \sum_{j=1}^{M} G_j^T(k+1) A_j^{+T} ,$$

and finally

$$\bar{S}(k+1) = \left[I - \sum_{j=1}^{M} A_j^+ G_j(k+1)H\right] \bar{\Phi}\, \bar{S}(k) \bar{\Phi}^T \left[I - \sum_{j=1}^{M} A_j^+ G_j(k+1)H\right]^T$$

$$+\left[I - \sum_{j=1}^{M} A_j^+ G_j(k+1)H\right] Q \left[I - \sum_{j=1}^{M} A_j^+ G_j(k+1)H\right]^T$$

$$+ \sum_{j=1}^{M} A_j^+ G_j(k+1) R \sum_{j=1}^{M} G_j^T(k+1) A_j^{+T} . \qquad (4.21)$$

Eq. (4.21) represents an iterative expression for the computation of the theoretical sub-optimal covariance of

estimation error. To initialize the computation of $\bar{S}(k+1)$, the initial value of $\bar{S}(0)$ should be chosen as

$$\bar{S}(0) = E\, \underline{\bar{e}}(0)\underline{\bar{e}}^T(0)$$

$$= E\left[\underline{x}(0) - \sum_{j=1}^{M} A_j^+ \hat{\underline{y}}_j(0/-1)\right]^2$$

$$= E\left[\underline{x}(0) - \bar{\underline{x}}(0/-1)\right]^2$$

$$\doteq P(0/-1) \ .$$

The discrete matrix expression for the theoretical sub-optimal covariance of estimation error $\bar{S}(k)$, may be used as a reference to evaluate the performance of the reduced set of filters, i.e., how close to the "true" values are the sub-optimal estimates. The solution for $\bar{S}(k)$ should be compared with the experimental variance of sub-optimal estimates.

C. EXAMPLE 2

The model used to investigate and to illustrate the technique presented in this chapter, was exactly the same fourth-order plant used in Example 1, subjected to the same deterministic and random forcing functions and with precisely the same measurements. The continuous system was discretized using the same sampling time T = 0.05 sec.,

and therefore the same $\bar{\Phi}$ and Γ matrices were obtained, arriving at the discretized system described by

$$\underline{x}(k+1) = \bar{\Phi}\underline{x}(k) + \Gamma \underline{u}(k) + \underline{w}(k) \quad , \tag{2.7}$$

with measurement

$$\underline{z}(k) = H\underline{x}(k) + \underline{v}(k) \quad .$$

The goal of this example, as in Example 1, was to estimate only two states of the fourth-order system (x_1 and x_2) by means of a reduced-order filter. Applying the technique discussed in Section A of this chapter, the original model (4^{th}order) could be partitioned into two second-order models, but in order to make the computational procedures much simpler, the original model was partitioned into four scalar models. This partitioning approach facilitated the selection of the A_i matrices and the computation of their respective pseudo-inverses, A_i^+.

The solution to the partitioning problem was achieved by trial and error. Four new vectors $\underline{y}_i$, i = 1,2,3,4 were defined by

$$y_i(k) = A_i\underline{x}(k) \quad , \quad i = 1,2,3,4.$$

The four A_i (1 x 4) matrices selected were

$$A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$$

It is important to remember here that all the A_i matrices should have the same dimensions, and that the choice of the A_i's in this case renders the y_i's scalar quantities.

The selection of the A_i matrices was made by trial and error (in this case obvious and simple) in order to get the values of proper A_i^+'s [1] that would fulfill the requirement established by Eq. (4.3), in this particular case

$$\sum_{i=1}^{4} A_i^+ A_i = I .$$

The pseudo-inverse for each A_i matrix was found using Eq. (4.2)

$$A_i \, A_i^+ \, A_i = A_i \, , \qquad (4.2)$$

---

[1] Werther (Ref.9) presents a computation method that yields to matrix inversion or pseudo-inversion.

from which the four A_i^+ (4 x 1) matrices were found to be[1]

$$A_1^+ = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad A_2^+ = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \qquad A_3^+ = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \qquad A_4^+ = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

The partitioning of the original state vector is then summarized by

$$y_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \underline{x}(k) = x_1(k)$$

$$y_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \underline{x}(k) = x_2(k)$$

$$y_3 = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \underline{x}(k) = x_3(k)$$

$$y_4 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \underline{x}(k) = x_4(k)$$

with dynamics as specified by Eq. (4.6)

This partitioning can be interpreted as an apparent canonization of the fourth-order model into four first-order models, represented each one by one state of the original system.

Having selected the A_i and A_i^+ matrices, four coupled scalar filters were implemented in parallel using

---

[1] It can be easily verified, that the selection of the A_i and A_i^+ matrices agrees with Eqs. (4.2) and (4.3).

Eqs. (4.13) and (4.14), but due to the presence of a deterministic forcing function, Eq. (4.13) was changed slightly as follows,

$$\hat{y}_i(k/k) = A_i\bar{\Phi}\sum_{j=1}^{4} A_i^+\hat{y}_i(k-1/k-1)+\Gamma_i u(k-1)+G_i(k)\left[z(k)\right.$$

$$\left.-H\sum_{j=1}^{4} A_j^+\left[\hat{y}_j(k/k-1) + \Gamma_j u(k-1)\right]\right] \quad i=1,2,3,4 \tag{4.22}$$

to show the effect of u(k) on the prediction step.

The following dynamics were used for the computation of the filter gains and theoretical covariances.

$$\bar{\Phi}_i = A_i\bar{\Phi}A_i^+ \qquad i = 1,2,3,4$$

$$\phi_1 = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\phi_{11} & \cdots & \cdots & \phi_{14}\\ \vdots & \phi_{22} & & \vdots\\ \vdots & & \phi_{33} & \vdots\\ \phi_{41} & \cdots & \cdots & \phi_{44}\end{bmatrix}\begin{bmatrix}1\\0\\0\\0\end{bmatrix}$$

$$\Gamma_1 = \Gamma_{11} ,$$

where ℓ_1 and ℓ_{11} are scalars. Similarly,

$$\ell_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \ell_{11} & \cdots & \cdots & \ell_{14} \\ \vdots & \ell_{22} & & \vdots \\ \vdots & & \ell_{33} & \vdots \\ \ell_{41} & \cdots & \cdots & \ell_{44} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\ell_2 = \ell_{22} ,$$

$$\ell_3 = \ell_{33} ,$$

and

$$\ell_4 = \ell_{44} .$$

Also,

$$\Gamma_i = A_i \Gamma \qquad i = 1,2,3,4$$

where

$$\gamma_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \gamma_{11} \\ \gamma_{21} \\ \gamma_{31} \\ \gamma_{41} \end{bmatrix} = \gamma_{11}$$

and similarly,

$$\gamma_2 = \gamma_{21}$$
$$\gamma_3 = \gamma_{31}$$

and

$$\gamma_4 = \gamma_{41} \cdot$$

Note that the symbols ϕ_1, ϕ_2, ... γ_1, γ_2 ... etc., correspond to scalar dynamics of the reduced processes, whereas the symbols ϕ_{11}, ϕ_{22}, ... γ_{11}, γ_{21}, ... etc., correspond to scalar elements of the dynamics of the full system.

The Q matrix was partitioned as follows:

$$Q_i = A_i Q A_i^+ \qquad i = 1,2,3,4.$$

then

$$q_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_{11} & \cdot & \cdot & \cdot & q_{14} \\ \cdot & & 22 & & \\ \cdot & & & 33 & \\ \cdot & & & & \\ q_{41} & & & & 44 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$q_1 = q_{11} ,$$

and similarly

$$q_2 = q_{22} ,$$

$$q_3 = q_{33} ,$$

and

$$q_4 = q_{44} ,$$

where q_1, q_2 ... q_{11}, q_{22} ... etc., represent scalar quantities.

The measurement process was specified by

$$z(k) = H \sum_{j=1}^{4} A_j^+ y_j(k) + v(k)$$

with the measurement matrix $H = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$, and $v(k) : N(0.0, 0.25)$.

For gains and variance calculations for each filter, the measurement matrix H(1 x 4) was partitioned as follows:

$$H_i = HA_i^+ \qquad i = 1,2,3,4$$

from which

$$h_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

or

$$h_1 = 1,$$

$$h_2 = 0,$$

$$h_3 = 0,$$

and

$$h_4 = 0,$$

with h_1, h_2 ... being scalars.

The initialization values for the reduced filters were computed using Eqs. (4.15) and (4.16), from which the initial guesses for the covariance of estimation error $P_i(0/-1)$ and expected value of the state vector $\hat{y}_i(0/-1)$ were obtained as follows:

$$P_i(0/-1) = A_i P(0/-1) A_i^+ \qquad i = 1,2,3,4$$

which gives

$$p_1(0/-1) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_{11}(0/-1) & & & \\ & p_{22}(0/-1) & & \\ & & p_{33}(0/-1) & \\ & & & p_{44}(0/-1) \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

or

$$p_1(0/-1) = p_{11}(0/-1) = 0.5 ,$$

$$p_2(0/-1) = p_{22}(0/-1) = 2.0 ,$$

$$p_3(0/-1) = p_{33}(0/-1) = 2.0 ,$$

and

$$p_4(0/-1) = p_{44}(0/-1) = 4.0 ,$$

where $p_1(0/-1)$ ..., $p_{11}(0/-1)$ ..., are scalars, with $p_1(0/-1)$, $p_2(0/-1)$, ... etc., representing the initial values of covariance for the reduced filters.

For the initial values of the estimates it is easily seen that

$$\hat{y}_1(0/-1) = \hat{x}_1(0/-1) = 0.0 ,$$

$$\hat{y}_2(0/-1) = \hat{x}_2(0/-1) = 0.0 ,$$

$$\hat{y}_3(0/-1) = \hat{x}_3(0/-1) = 0.0 ,$$

and

$$\hat{y}_4(0/-1) = \hat{x}_4(0/-1) = 0.0$$

will provide the filters with unbiased initial values.

Then four scalar, discrete, sub-optimal filters, in this example coupled[1] only through the measurement process. were implemented and run in parallel in accordance with Eqs. (4.22), (4.14), and (4.7). At the end of each sampling interval the outputs from the four filters were combined by the expression

$$\hat{\underline{x}}(k/k) = \sum_{j=1}^{4} A_j^{+} \hat{y}_j(k/k) \ ,$$

in order to recover the sub-optimal estimates of the original vector $\underline{x}(k)$, from which only the estimation of x_1 and x_2 was of particular interest.

At the same time, the theoretical sub-optimal covariance of estimation error $\bar{S}(k)$, was computed using recursive matrix Eq. (4.21), in order to evaluate the selection of the set of constant matrices A_i. This is possible to see when $\bar{S}(k)$ is compared with the experimental reduced covariance of estimation error.

Again a Monte Carlo simulation[2] was performed with 200 replications of the trajectory, each trajectory made

---

1 This is not generally so. The uncoupling here is a consequence of the choice of the A_i matrices. The y_i state equations need not be uncoupled, but the associated gain and variance equations are, however, always uncoupled.

2 As in Example 1, a computer System 360/67 was used for the simulation and solution of this problem. See Appendix C for corresponding computer program.

of 50 samples, with sampling intervals of 0.05 seconds. As a result of this simulation, ensemble values for the mean and experimental covariance of estimation error, as a function of time, were obtained in x_1 and x_2 for the optimal and sub-optimal filters as a means of comparing their performance.

Numerical results of this simulation are tabulated in Tables (IV-2) through (IV-6) and graphical representation of the results is shown in Figs. (4.3) through (4.6).

Table IV-1

PHI MATRIX

0.9999	0.0499	0.0012	0.0000
-0.0066	0.9946	0.0480	0.0009
-0.3644	-0.2981	0.8871	0.0307
-12.2713	-10.1815	-3.9181	0.3043

GAMMA MATRIX

0.0001
0.0066
0.3644
12.2713

Q MATRIX

0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0008
0.0000	0.0000	0.0013	0.0447
0.0000	0.0008	0.0447	1.5059

VARIANCE OF FORCING NOISE = 0.01

VARIANCE OF MEASUREMENT NOISE = 0.25

INITIALIZATION - P(0/-1)

0.5000	0.0	0.0	0.0
0.0	2.0000	0.0	0.0
0.0	0.0	2.0000	0.0
0.0	0.0	0.0	4.0000

INITIALIZATION - X(0/-1)

0.0
0.0
0.0
0.0

Table IV-2

GAINS FOR FULL AND REDUCED FILTERS

K	G(1,1)	G(2,1)	G(3,1)	G(4,1)	G1(1,1)	G2(1,1)
1	0.6667	0.0	0.0	0.0	0.6667	0.0
2	0.4071	0.2333	-0.2095	-7.2845	0.4000	0.0
3	0.3097	0.4075	-0.6537	-9.4633	0.2857	0.0
4	0.2677	0.5088	-1.1719	-9.0573	0.2222	0.0
5	0.2477	0.5314	-1.6080	-7.0288	0.1817	0.0
6	0.2349	0.4901	-1.8640	-4.4015	0.1538	0.0
7	0.2227	0.4108	-1.9266	-1.9563	0.1333	0.0
8	0.2087	0.3175	-1.8378	-0.0721	0.1176	0.0
9	0.1928	0.2268	-1.6557	1.1851	0.1052	0.0
10	0.1758	0.1474	-1.4305	1.9077	0.0952	0.0
11	0.1584	0.0825	-1.1970	2.2348	0.0869	0.0
12	0.1412	0.0322	-0.9760	2.2961	0.0799	0.0
13	0.1247	-0.0051	-0.7780	2.1931	0.0740	0.0
14	0.1093	-0.0315	-0.6072	1.9984	0.0689	0.0
15	0.0950	-0.0490	-0.4637	1.7610	0.0644	0.0
16	0.0819	-0.0596	-0.3457	1.5120	0.0605	0.0
17	0.0701	-0.0647	-0.2505	1.2705	0.0571	0.0
18	0.0596	-0.0660	-0.1752	1.0474	0.0540	0.0
19	0.0502	-0.0643	-0.1167	0.8480	0.0512	0.0
20	0.0420	-0.0606	-0.0723	0.6746	0.0487	0.0
21	0.0348	-0.0556	-0.0395	0.5269	0.0464	0.0
22	0.0286	-0.0499	-0.0161	0.4036	0.0444	0.0
23	0.0233	-0.0438	-0.0001	0.3026	0.0425	0.0
24	0.0188	-0.0378	0.0100	0.2215	0.0407	0.0
25	0.0151	-0.0320	0.0157	0.1576	0.0391	0.0
26	0.0120	-0.0266	0.0181	0.1084	0.0377	0.0
27	0.0095	-0.0217	0.0181	0.0714	0.0363	0.0
28	0.0075	-0.0174	0.0166	0.0444	0.0350	0.0
29	0.0059	-0.0136	0.0141	0.0254	0.0338	0.0
30	0.0047	-0.0105	0.0111	0.0127	0.0327	0.0
31	0.0037	-0.0078	0.0080	0.0046	0.0317	0.0
32	0.0031	-0.0057	0.0050	-0.0000	0.0307	0.0
33	0.0026	-0.0041	0.0023	-0.0022	0.0298	0.0
34	0.0022	-0.0028	-0.0001	-0.0028	0.0289	0.0
35	0.0020	-0.0018	-0.0022	-0.0024	0.0281	0.0
36	0.0018	-0.0011	-0.0038	-0.0015	0.0273	0.0
37	0.0018	-0.0006	-0.0050	-0.0004	0.0266	0.0
38	0.0017	-0.0003	-0.0060	0.0007	0.0259	0.0
39	0.0017	-0.0001	-0.0066	0.0017	0.0252	0.0
40	0.0017	0.0000	-0.0070	0.0024	0.0246	0.0
41	0.0017	0.0001	-0.0073	0.0030	0.0240	0.0
42	0.0017	0.0001	-0.0074	0.0033	0.0234	0.0
43	0.0017	0.0001	-0.0074	0.0034	0.0229	0.0
44	0.0017	0.0001	-0.0073	0.0033	0.0224	0.0
45	0.0017	0.0000	-0.0072	0.0031	0.0219	0.0
46	0.0017	0.0000	-0.0071	0.0028	0.0214	0.0
47	0.0017	-0.0000	-0.0070	0.0025	0.0210	0.0
48	0.0017	-0.0000	-0.0068	0.0022	0.0205	0.0
49	0.0017	-0.0001	-0.0067	0.0018	0.0201	0.0
50	0.0017	-0.0001	-0.0066	0.0015	0.0197	0.0

G (4x1) represents the matrix of gains for the optimal filter.

G1 and G2 represent the gains used in the sub-optimal filters 1 and 2.

Table IV-3

THEORETICAL VARIANCE OF ESTIMATION ERROR
FULL PLANT

K	PK(1,1)	PK(2,2)	PK(3,3)	PK(4,4)
1	0.1667	2.0000	2.0000	4.0000
2	0.1018	1.9603	1.7605	242.6279
3	0.0774	1.8002	2.3189	346.3362
4	0.0669	1.5182	3.9792	261.1479
5	0.0619	1.1771	5.7036	142.2309
6	0.0587	0.8478	6.6569	60.6997
7	0.0557	0.5745	6.6825	21.3209
8	0.0522	0.3709	6.0419	8.2223
9	0.0482	0.2307	5.0782	6.9855
10	0.0439	0.1400	4.0514	9.3412
11	0.0396	0.0847	3.1097	11.6978
12	0.0353	0.0529	2.3160	12.9672
13	0.0312	0.0361	1.6824	13.1242
14	0.0273	0.0282	1.1960	12.4778
15	0.0237	0.0255	0.8337	11.3650
16	0.0205	0.0254	0.5710	10.0516
17	0.0175	0.0264	0.3855	8.7171
18	0.0149	0.0275	0.2581	7.4696
19	0.0126	0.0283	0.1736	6.3652
20	0.0105	0.0286	0.1199	5.4260
21	0.0087	0.0282	0.0877	4.6526
22	0.0072	0.0273	0.0701	4.0336
23	0.0058	0.0259	0.0619	3.5512
24	0.0047	0.0241	0.0595	3.1854
25	0.0038	0.0221	0.0601	2.9159
26	0.0030	0.0199	0.0619	2.7237
27	0.0024	0.0177	0.0639	2.5918
28	0.0019	0.0155	0.0654	2.5057
29	0.0015	0.0133	0.0659	2.4529
30	0.0012	0.0114	0.0654	2.4236
31	0.0009	0.0096	0.0640	2.4099
32	0.0008	0.0081	0.0618	2.4059
33	0.0006	0.0067	0.0590	2.4071
34	0.0006	0.0056	0.0557	2.4105
35	0.0005	0.0046	0.0523	2.4141
36	0.0005	0.0039	0.0489	2.4167
37	0.0004	0.0033	0.0455	2.4177
38	0.0004	0.0028	0.0424	2.4170
39	0.0004	0.0024	0.0396	2.4146
40	0.0004	0.0022	0.0371	2.4109
41	0.0004	0.0020	0.0349	2.4062
42	0.0004	0.0018	0.0330	2.4008
43	0.0004	0.0017	0.0315	2.3951
44	0.0004	0.0017	0.0303	2.3894
45	0.0004	0.0017	0.0293	2.3839
46	0.0004	0.0016	0.0285	2.3787
47	0.0004	0.0016	0.0279	2.3741
48	0.0004	0.0016	0.0275	2.3700
49	0.0004	0.0016	0.0272	2.3664
50	0.0004	0.0016	0.0270	2.3635

PK represents the diagonal elements of the theoretical covariance matrix for the optimal filter.

Table IV-4

EXPERIMENTAL VARIANCE OF ESTIMATION ERROR
FULL PLANT

K	VK(1,1)	VK(2,2)	VK(3,3)	VK(4,4)
1	0.1317	1.9066	1.9733	4.0880
2	0.0910	1.8538	1.7402	225.0217
3	0.0739	1.7340	2.2403	326.4661
4	0.0623	1.4891	3.7969	257.7678
5	0.0645	1.1562	5.6106	138.7382
6	0.0518	0.8086	5.9918	60.5067
7	0.0510	0.5760	6.5021	23.3325
8	0.0485	0.3807	6.0129	8.6785
9	0.0465	0.2391	5.1270	7.2472
10	0.0459	0.1500	4.3535	9.5983
11	0.0424	0.0927	3.5224	12.3814
12	0.0394	0.0557	2.7024	14.0400
13	0.0367	0.0360	1.9633	13.9882
14	0.0295	0.0276	1.3043	13.4596
15	0.0264	0.0240	0.9568	11.7134
16	0.0225	0.0239	0.6456	12.1689
17	0.0187	0.0256	0.4017	9.6542
18	0.0152	0.0259	0.2554	7.2280
19	0.0130	0.0271	0.1698	5.9514
20	0.0111	0.0269	0.1194	5.5124
21	0.0093	0.0275	0.0835	5.0569
22	0.0073	0.0253	0.0671	4.1066
23	0.0060	0.0241	0.0626	3.5153
24	0.0047	0.0218	0.0616	2.8292
25	0.0039	0.0203	0.0586	3.1777
26	0.0031	0.0186	0.0564	2.4777
27	0.0025	0.0166	0.0546	2.7436
28	0.0020	0.0150	0.0546	2.7655
29	0.0016	0.0131	0.0550	2.6672
30	0.0012	0.0115	0.0581	2.0752
31	0.0010	0.0099	0.0586	2.4300
32	0.0008	0.0084	0.0572	2.7947
33	0.0007	0.0072	0.0546	2.5196
34	0.0006	0.0060	0.0495	2.1787
35	0.0005	0.0051	0.0437	2.5699
36	0.0004	0.0043	0.0422	2.2536
37	0.0004	0.0035	0.0438	2.5338
38	0.0004	0.0029	0.0449	2.3031
39	0.0004	0.0025	0.0427	2.0479
40	0.0004	0.0021	0.0373	2.4109
41	0.0004	0.0019	0.0308	2.2848
42	0.0004	0.0018	0.0261	2.5309
43	0.0004	0.0016	0.0242	2.2161
44	0.0004	0.0015	0.0261	2.4027
45	0.0004	0.0014	0.0299	1.9605
46	0.0004	0.0014	0.0316	1.9165
47	0.0004	0.0014	0.0306	2.2086
48	0.0004	0.0014	0.0297	2.3207
49	0.0004	0.0015	0.0281	2.1990
50	0.0004	0.0016	0.0229	2.8940

VK represents the diagonal elements of the experimental covariance matrix for the optimal process.

Table IV-5

SUBOPTIMAL VARIANCE OF ESTIMATION ERROR

K	SR(1,1)	SR(2,2)	VR(1,1)	VR(2,2)
1	0.1667	2.0000	0.1308	1.9041
2	0.1018	1.9833	0.0899	1.8888
3	0.0777	1.8799	0.0725	1.7924
4	0.0682	1.6698	0.0615	1.5970
5	0.0653	1.3827	0.0627	1.3288
6	0.0653	1.0666	0.0578	1.0320
7	0.0661	0.7655	0.0603	0.7471
8	0.0662	0.5098	0.0610	0.5028
9	0.0650	0.3136	0.0603	0.3134
10	0.0622	0.1780	0.0598	0.1806
11	0.0579	0.0954	0.0553	0.0980
12	0.0525	0.0538	0.0532	0.0549
13	0.0464	0.0406	0.0490	0.0393
14	0.0399	0.0443	0.0406	0.0407
15	0.0336	0.0558	0.0351	0.0506
16	0.0277	0.0687	0.0294	0.0627
17	0.0225	0.0791	0.0241	0.0729
18	0.0180	0.0852	0.0187	0.0793
19	0.0143	0.0864	0.0149	0.0813
20	0.0114	0.0834	0.0120	0.0794
21	0.0091	0.0770	0.0096	0.0743
22	0.0075	0.0685	0.0072	0.0669
23	0.0063	0.0589	0.0061	0.0582
24	0.0055	0.0492	0.0048	0.0493
25	0.0049	0.0401	0.0046	0.0406
26	0.0046	0.0320	0.0041	0.0327
27	0.0044	0.0251	0.0038	0.0258
28	0.0042	0.0196	0.0040	0.0202
29	0.0041	0.0153	0.0039	0.0157
30	0.0039	0.0121	0.0038	0.0123
31	0.0038	0.0099	0.0036	0.0099
32	0.0037	0.0084	0.0035	0.0083
33	0.0035	0.0074	0.0034	0.0074
34	0.0033	0.0068	0.0030	0.0069
35	0.0032	0.0065	0.0028	0.0066
36	0.0030	0.0063	0.0026	0.0063
37	0.0028	0.0062	0.0025	0.0061
38	0.0026	0.0061	0.0025	0.0058
39	0.0025	0.0060	0.0026	0.0056
40	0.0023	0.0058	0.0026	0.0054
41	0.0022	0.0056	0.0025	0.0052
42	0.0021	0.0054	0.0023	0.0050
43	0.0020	0.0052	0.0023	0.0047
44	0.0019	0.0050	0.0022	0.0045
45	0.0018	0.0047	0.0020	0.0042
46	0.0017	0.0045	0.0020	0.0040
47	0.0016	0.0043	0.0018	0.0039
48	0.0016	0.0041	0.0017	0.0038
49	0.0015	0.0039	0.0017	0.0038
50	0.0015	0.0037	0.0017	0.0038

SR represents two of the diagonal elements of the sub-optimal theoretical covariance of estimation error, corresponding to the sub-optimal estimation of x_1 and x_2.

VR follows exactly the same pattern as SR, but it refers to the experimental covariance of estimation error.

Table IV-6

EXPERIMENTAL MEANS OF ESTIMATION ERROR

K	MEANE1	MEANE2	MEANE3	MEANE4	MEANE1R	MEANE2R
1	0.0312	0.0500	-0.0350	-0.1543	0.0312	0.0500
2	0.0263	0.0436	-0.0543	-0.5355	0.0264	0.0478
3	0.0282	0.0398	-0.0893	-0.7945	0.0285	0.0439
4	0.0330	0.0398	-0.1411	-0.8346	0.0328	0.0379
5	0.0362	0.0349	-0.1811	-0.4616	0.0356	0.0301
6	0.0313	0.0120	-0.1482	-0.1316	0.0328	0.0213
7	0.0289	-0.0006	-0.1229	0.1844	0.0317	0.0120
8	0.0041	-0.0439	0.1008	0.1273	0.0178	0.0030
9	-0.0014	-0.0429	0.1368	0.0809	0.0142	-0.0054
10	0.0055	-0.0284	0.0687	0.1810	0.0170	-0.0124
11	0.0150	-0.0193	-0.0130	0.0214	0.0211	-0.0176
12	0.0031	-0.0223	0.0684	-0.0040	0.0134	-0.0213
13	-0.0017	-0.0188	0.0900	-0.1130	0.0093	-0.0238
14	-0.0060	-0.0134	0.1014	-0.2676	0.0052	-0.0251
15	-0.0075	-0.0083	0.0901	-0.3582	0.0026	-0.0254
16	-0.0046	-0.0065	0.0577	-0.3345	0.0031	-0.0250
17	-0.0016	-0.0069	0.0383	0.0395	0.0042	-0.0239
18	0.0031	-0.0106	0.0206	-0.0433	0.0073	-0.0220
19	0.0066	-0.0148	0.0064	-0.0677	0.0101	-0.0198
20	0.0054	-0.0139	0.0036	-0.0764	0.0085	-0.0179
21	0.0050	-0.0140	0.0084	0.2286	0.0078	-0.0164
22	0.0024	-0.0100	0.0190	0.1291	0.0040	-0.0148
23	0.0030	-0.0111	0.0238	0.0811	0.0053	-0.0131
24	0.0025	-0.0099	0.0236	-0.0649	0.0046	-0.0115
25	0.0018	-0.0082	0.0217	-0.0134	0.0033	-0.0102
26	0.0015	-0.0074	0.0204	-0.0375	0.0030	-0.0090
27	0.0013	-0.0068	0.0202	0.0076	0.0032	-0.0080
28	0.0011	-0.0059	0.0262	0.1873	0.0034	-0.0069
29	0.0006	-0.0038	0.0332	0.1184	0.0015	-0.0055
30	0.0006	-0.0025	0.0299	-0.2134	0.0022	-0.0040
31	0.0007	-0.0016	0.0225	-0.1106	0.0036	-0.0028
32	0.0007	-0.0008	0.0181	-0.0725	0.0043	-0.0019
33	0.0008	-0.0001	0.0093	-0.2493	0.0044	-0.0013
34	0.0007	0.0003	0.0026	-0.0454	0.0033	-0.0013
35	0.0006	0.0005	0.0036	0.0645	0.0020	-0.0014
36	0.0007	0.0006	0.0030	-0.0675	0.0020	-0.0015
37	0.0007	0.0007	0.0003	-0.0409	0.0020	-0.0016
38	0.0007	0.0006	-0.0055	-0.1695	0.0014	-0.0019
39	0.0007	0.0002	-0.0080	0.0391	0.0012	-0.0024
40	0.0007	-0.0001	-0.0045	0.0863	0.0005	-0.0027
41	0.0008	-0.0002	0.0006	0.1190	0.0015	-0.0027
42	0.0007	-0.0001	0.0038	0.0133	0.0005	-0.0023
43	0.0008	0.0002	0.0048	0.0356	0.0015	-0.0019
44	0.0008	0.0004	0.0035	-0.0646	0.0019	-0.0014
45	0.0008	0.0005	0.0012	-0.0425	0.0009	-0.0010
46	0.0008	0.0006	0.0043	0.1323	0.0006	-0.0007
47	0.0007	0.0009	0.0081	0.0145	-0.0009	-0.0002
48	0.0008	0.0013	0.0074	-0.0337	-0.0008	0.0003
49	0.0009	0.0017	0.0071	0.0159	-0.0005	0.0009
50	0.0010	0.0020	0.0057	-0.0529	0.0001	0.0015

MEANE1 through MEANE4 represent the mean of estimation error in the complete state vector by means of optimal filtering.

MEANE1R and MEANE2R represent the sub-optimal mean of estimation error in states x_1 and x_2 by means of reduced filters.

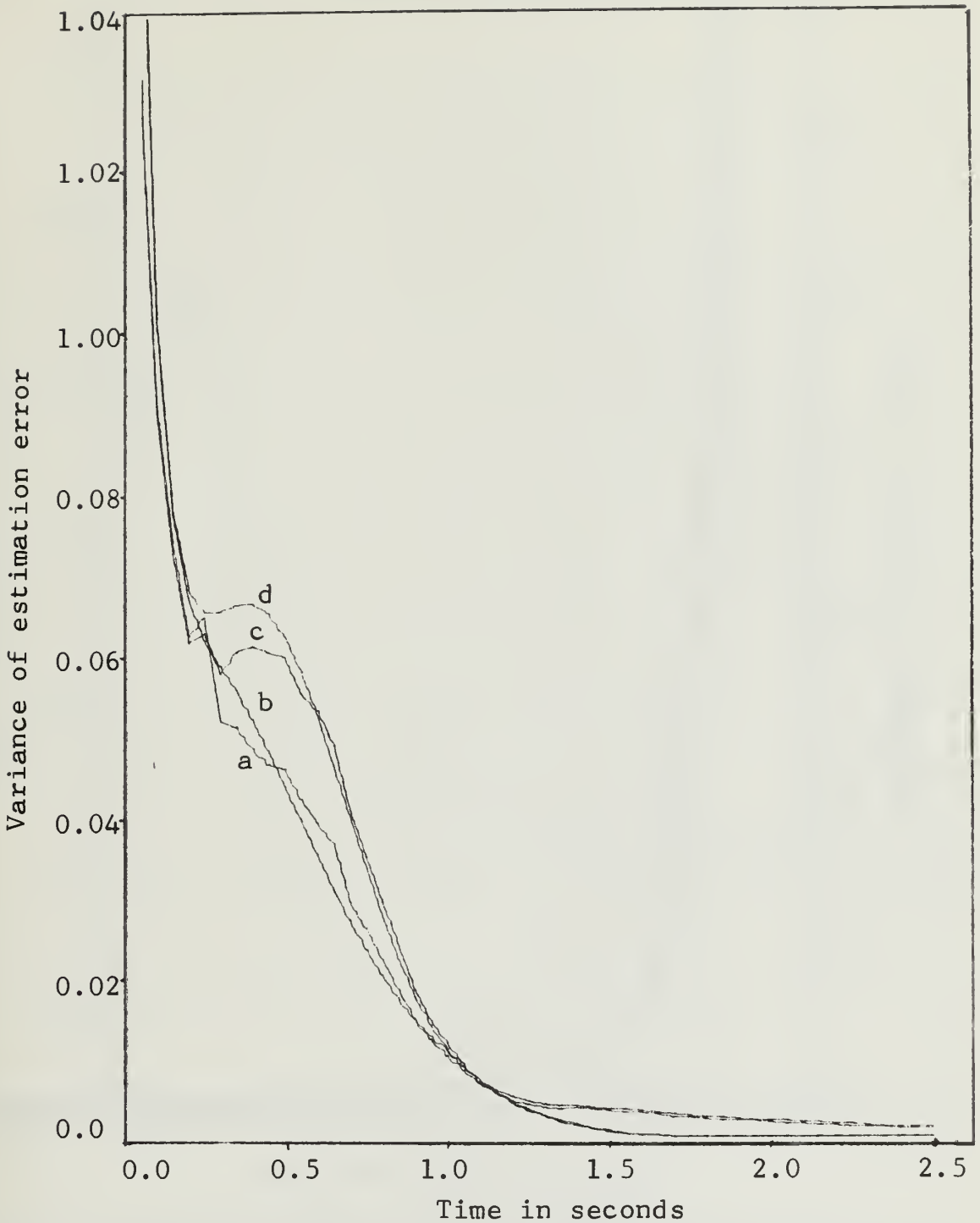


Fig. 4.3 - Graphical plot of the variance of estimation error in x_1 vs. time. Curves a and b show the experimental and theoretical variance of estimation error for the optimal process. Curves c and d show the sub-optimal experimental and theoretical covariance of estimation error.

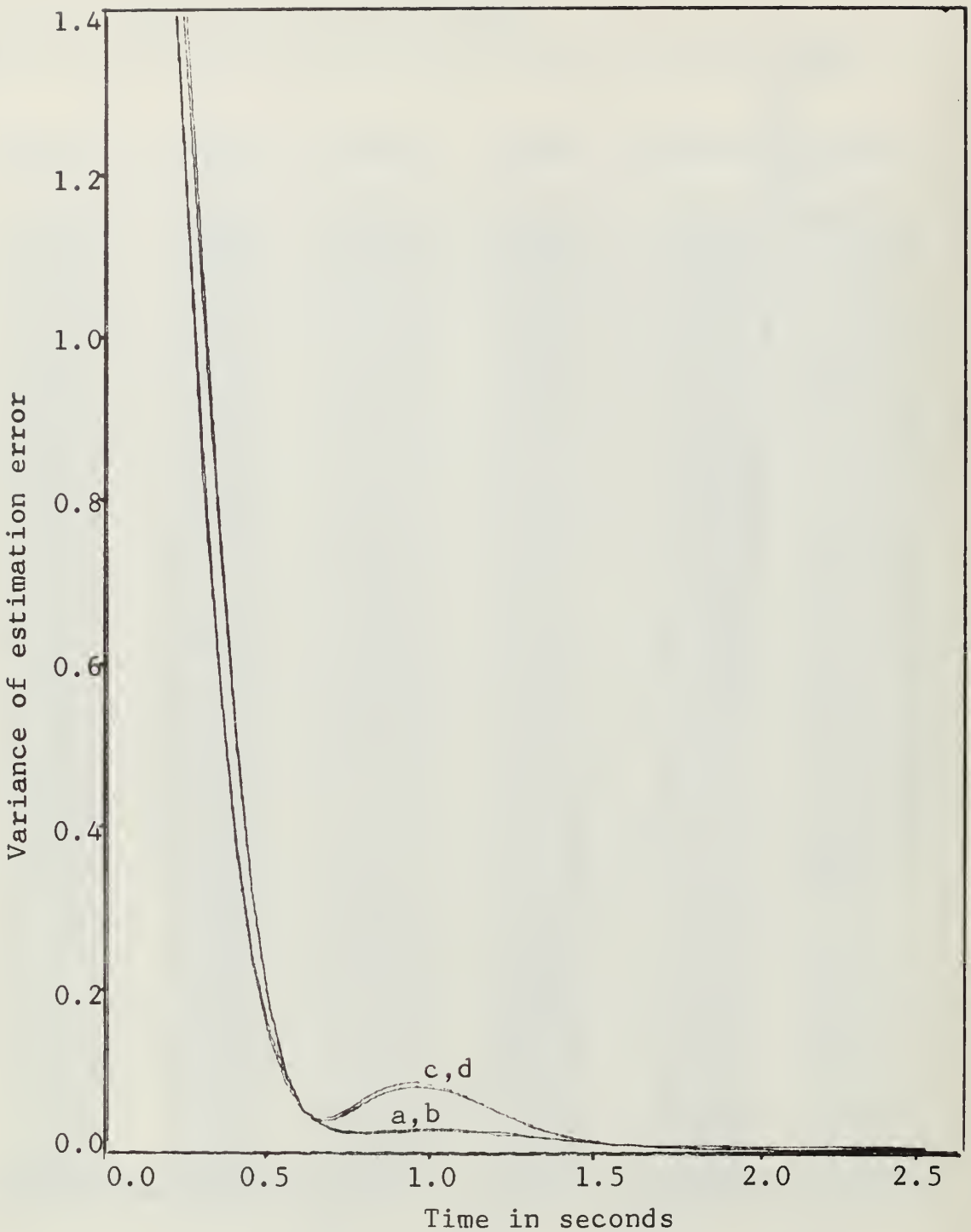


Fig. 4.4 - Graphical plot of the variance of estimation error in x_2 vs. time. Curves a and b show the experimental and theoretical variance of estimation error for the optimal process. Curves c and d show the sub-optimal case.

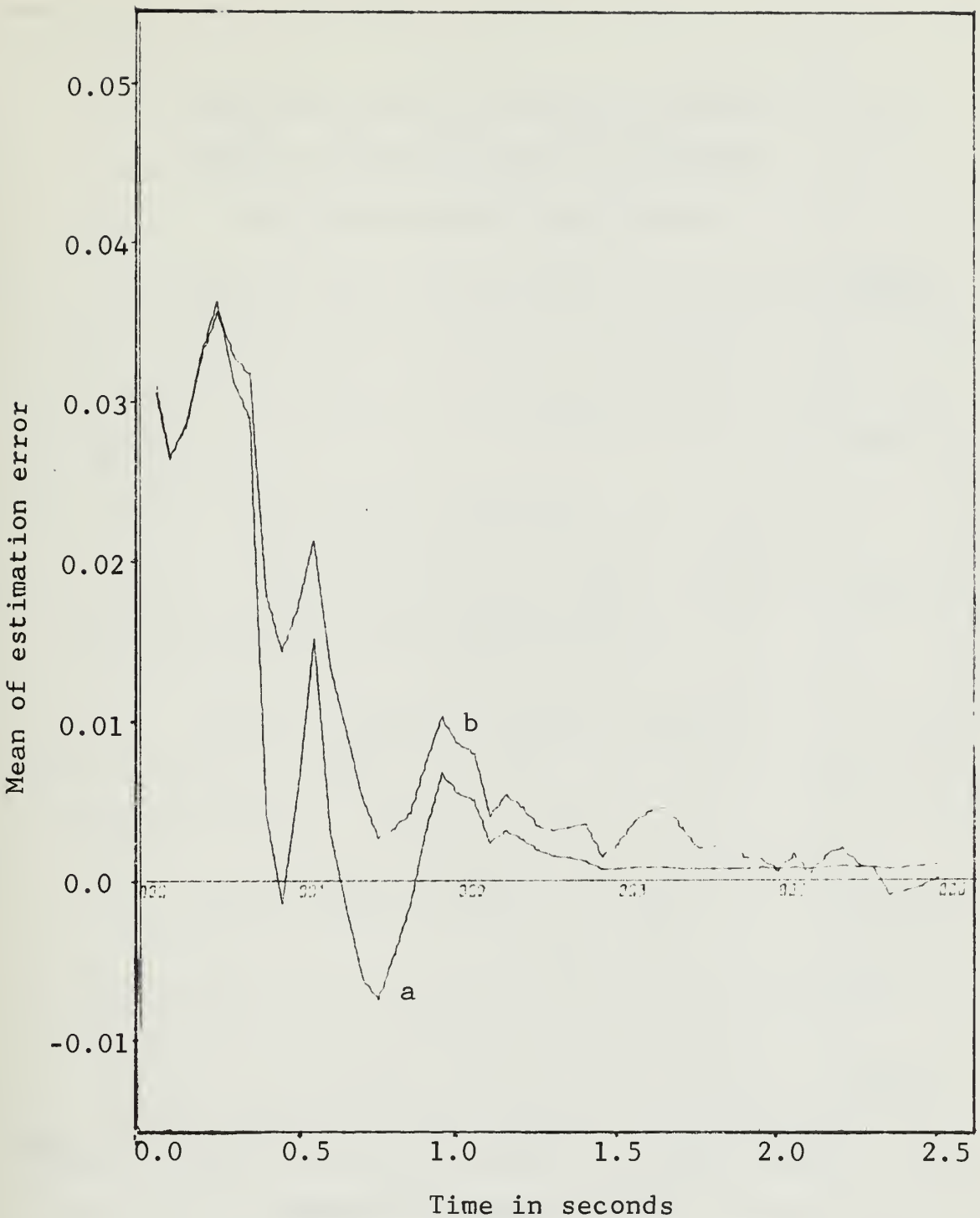


Fig. 4.5 - Graphical plot of the mean of estimation error in x_1 vs. time. Curve a shows the mean of estimation error for the optimal estimation process. Curve b shows that for the sub-optimal case.

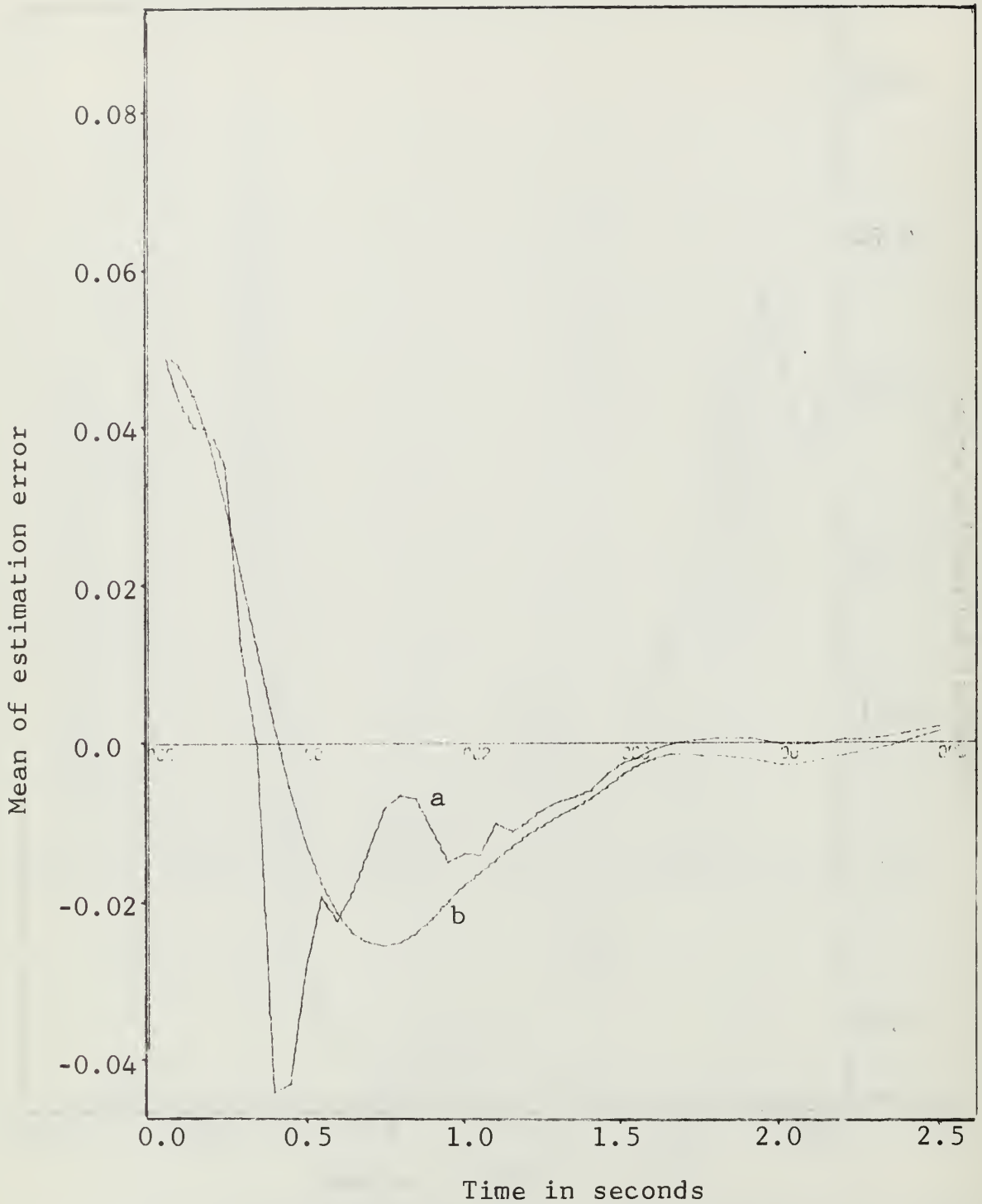


Fig. 4.6 - Graphical plot of the means of estimation error in x_2 vs. time. Curve a shows the mean of estimation error for the optimal estimation process. Curve b shows that for the sub-optimal case.

V. IMPLEMENTATION OF A REDUCED DISCRETE KALMAN FILTER BY USING THE DOMINANT ROOTS OF THE ORIGINAL PROCESS

The general idea of producing a reduced sub-optimal filter (Chapter III) or several sub-optimal filters of reduced order (Chapter IV), is presented here using a different approach.

A. THE DOMINANT-ROOTS APPROACH

It is assumed as in the previous chapters, that the full system dynamics are represented by

$$\underline{x}(k+1) = \bar{\Phi}\underline{x}(k) + \Gamma\left[\underline{u}(k) + \underline{w}(k)\right] , \qquad (2.9)$$

with measurement

$$\underline{z}(k) = H\underline{x}(k) + \underline{v}(k) , \qquad (2.3)$$

and with the random processes involved, $\underline{w}(k)$ and $\underline{v}(k)$, fully described by Eqs. (2.11) through (2.16).

It is clear that the order of the system is n and that the goal of the problem is to estimate only r states ($r<n$) of the process; therefore it will be necessary to produce a reduced model of the plant in order to use its dynamics in the implementation of the reduced filter.

It is common practice when handling classical automatic control problems to assume, or to require, that the high-order system be dominated by one pair of complex conjugate roots. This will normally permit the use of a second-order characteristic equation, and will simplify analysis and synthesis because the second-order system has been completely solved and all kind of curves, etc., are available. Applying the same idea to the n^{th}-order, linear plant under consideration, but without the constraint of a pure second-order approximation, the dominant-roots representation will be determined mainly by the number r of states to be estimated and by the amount of performance degradation introduced when choosing the reduced model.

It is noted that the number r of states to be estimated will specify the minimum number of roots which can be selected as dominant, but an upper limit in the selection of the dominant roots would be determined by the designer's judgement, based upon the amount of degradation introduced by the reduced model.

Due to the qualitative nature of the root-dominance criterion, it is very difficult to state a general procedure for the selection of the reduced-order model. Therefore a specific example will be used to illustrate the method.

Consider the fourth-order linear system with continuous dynamics used in Examples 1 and 2 and described by

$$\dot{\underline{x}}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -400 & -320 & -118 & -19 \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 400 \end{bmatrix} [u(t)+w(t)] \ , \tag{5.1}$$

with I.C

$$\underline{x}(0) = \underline{0} \ ,$$

and measurement

$$z(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \underline{x}(t) + v(t) \ .$$

The task of the problem is to estimate only two states of the original process (x_1 and x_2).

When solving Example 1 (Chapter 3), the whole process (Eq. 5.1) was discretized and then the dynamics of the discrete system were partitioned according to the number of states to be estimated. Here, instead of discretizing the whole system, the order of the continuous process will be lowered by proper choice of the dominant roots, and then the reduced continuous system will be discretized.

The system described by Eq. (5.1) may be rewritten as

$$\dot{x}_1(t) = x_2(t) \ , \tag{5.2a}$$

$$\dot{x}_2(t) = x_3(t) \ , \tag{5.2b}$$

$$\dot{x}_3(t) = x_4(t) \ , \tag{5.2c}$$

and

$$\dot{x}_4(t) = -400x_1(t)-320x_2(t)-118x_3(t)-19x_4(t)+400\ u(t)+w(t) \ . \tag{5.2d}$$

For convenience a new variable $\vartheta(t) = x_1(t)$ is defined; then Eqs. (5.2a,b,c) become

$$\dot{\vartheta}(t) = x_2(t) \ ,$$

$$\ddot{\vartheta}(t) = x_3(t) \ ,$$

$$\dddot{\vartheta}(t) = x_4(t) \ ,$$

and by direct substitution of these three equations into Eq. (5.2d) the latter becomes

$$\ddddot{\vartheta}(t) + 19\,\dddot{\vartheta}(t) + 118\,\ddot{\vartheta}(t) + 320\dot{\vartheta}(t) + 400\vartheta(t) = 400\left[u(t)+w(t)\right], \tag{5.3}$$

where

$$\ddddot{\vartheta}(t) = \frac{d^4\vartheta(t)}{dt^4} \ .$$

The dominant-root analysis of Eq. (5.3), which now fully represents the fourth-order system under consideration, will be made with the help of a pole-zero representation. This will require (as an easy approach) transformation of Eq. (5.3) into the s-domain, accomplished here by use of the Laplace transform. Then, considering all initial conditions equal to zero, Eq. (5.3) transforms into

$$\Theta(s)s^4+19\Theta(s)s^3+118\Theta(s)s^2+320\Theta(s)s+400\Theta(s)=400\left[U(s)+W(s)\right].$$

which after proper factorization gives

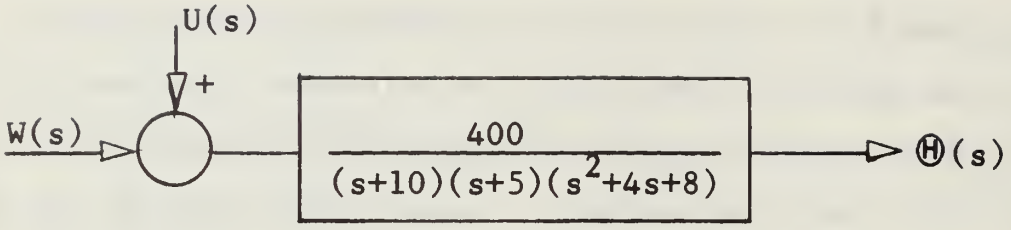
$$\Theta(s) = \frac{400\left[U(s)+W(s)\right]}{(s+10)(s+5)(s+2-j2)(s+2+j2)} ,$$

or

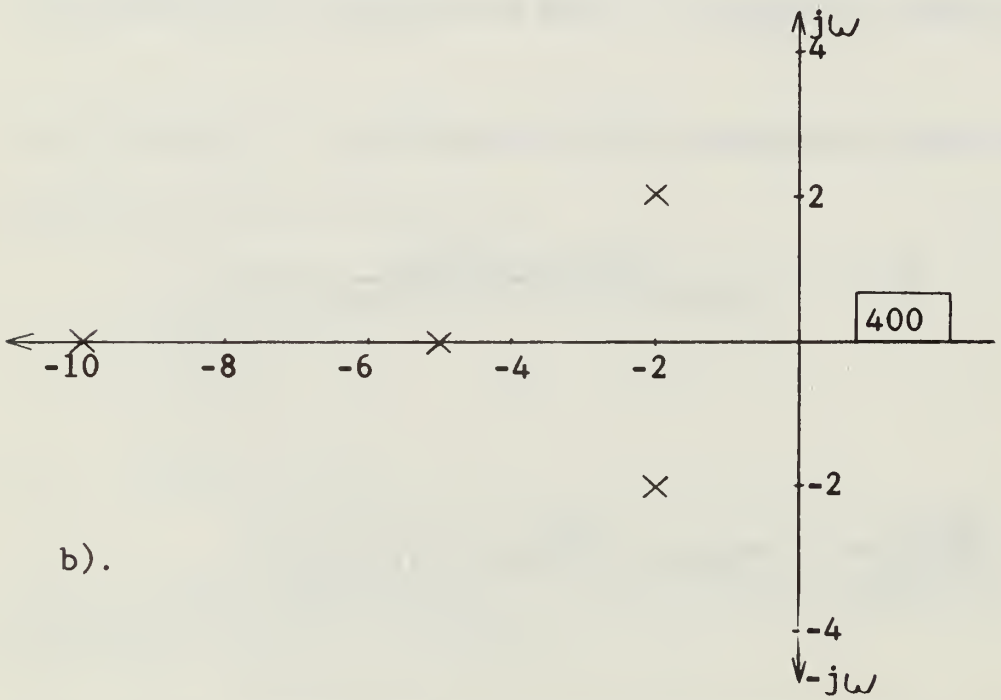
$$\frac{\Theta(s)}{U(s)+W(s)} = \frac{400}{(s+10)(s+5)(s+2-j2)(s+2+j2)} . \qquad (5.4)$$

Eq. (5.4), may be interpreted as the transfer function of the fourth-order system relating $\Theta(s)$ with the forcing functions (inputs).

A pole-zero representation of Eq. (5.4) (Fig.5.1b), shows a pair of complex conjugate roots located at $s = -2\pm j2$, and two real roots located at $s = -5$ and $s = -10$ respectively. If the dominance criterion is based upon the size of the real portion of the roots under consideration,



a).



b).

Fig. 5.1 - a). Block diagram representation of the process described by Eq. (5.4).

b). Pole-zero diagram for Eq. (5.4) indicating four roots at s = -5, s = -10, s = $-2\pm j2$ respectively, and with 400 considered to be the gain of the system.

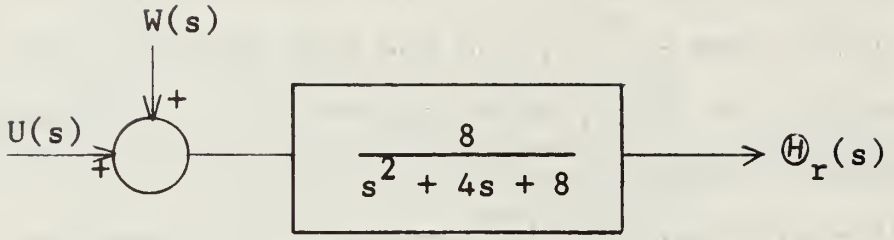
then it is clear for this particular example, that the less dominant root will be at s = -10 (this will give rise to a term of the form ϵ^{-10t}), and the most dominant roots will be those at s = $-2\pm j2$, with the root at s = -5 producing an intermediate effect.

As stated before, the purpose of this problem is to estimate only two states of the original system (x_1 and x_2). It therefore seems a good approach to select the roots at s = $-2\pm j2$ in order to produce a reduced second-order continuous system, whose dynamics will be used in the implementation of the reduced filter. It is important to note that the gain of the reduced system should be corrected in order to compensate for the disappearance of two of the roots of the original system. In this case the gain of the reduced system was made equal to the product of the magnitudes of the dominant roots, that is

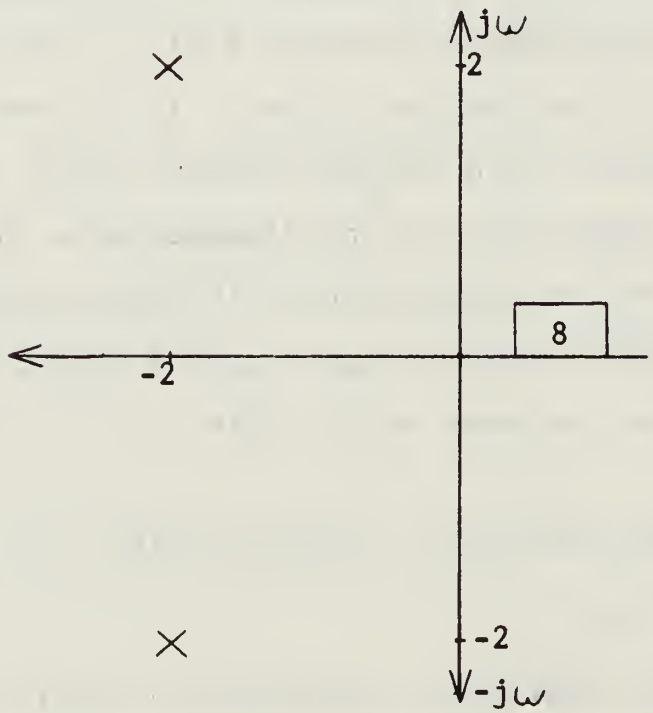
$$\text{gain reduction} = |-2-j2|\,|-2+j2| = 8.$$

See Fig. (5.2)

Another important consideration should be based upon the fact that the performance degradation introduced, and the number m of inputs (forcing functions) to the original system, will strongly determine the order of the reduced model when using dominant roots.



a).



b).

Fig. 5.2 - a). Block diagram representation of the reduced model.

b). Pole-zero diagram showing only the dominant roots. Gain of the system has been adjusted to 8.

After the selection of the pair of complex conjugate roots at $s = -2 \pm j2$, the reduced model may be expressed as

$$\frac{\Theta_r(s)}{U(s)+W(s)} = \frac{8}{(s+2-j2)(s+2+j2)} = \frac{8}{s^2+4s+8}$$

or transferring back to the differential equation form,

$$\ddot{\theta}_r(t) + 4\dot{\theta}_r(t) + 8\theta_r(t) = 8\left[u(t)+w(t)\right].$$

Recalling that $\theta_r(t) = x_1(t)$, the second-order reduced model becomes

$$\dot{x}_1(t) = x_2(t)$$

$$\dot{x}_2(t) = -8x_1(t) - 4x_2(t) + 8\left[u(t)+w(t)\right],$$

or

$$\dot{x}_r(t) = \begin{bmatrix} 0 & -1 \\ -8 & -4 \end{bmatrix} \underline{x}_r(t) + \begin{bmatrix} 0 \\ 8 \end{bmatrix} \left[u(t)+w(t)\right],$$

which represents the reduced model from which the reduced filter will be implemented (see Fig. 5.2).

No special derivation of the reduced-filter equations is necessary because the form of the equations will be exactly the same as those for the optimal filtering, and only the order will change.

The discrete reduced model is obtained from Eq. (5.5) using

$$\Phi_r = \epsilon^{A_r T} \quad , \tag{5.6}$$

and

$$\Gamma_r = \epsilon^{A_r T} \int_o^T \epsilon^{-A_r \tau} d\tau B_r \quad , \tag{5.7}$$

with $A_r(r \times r)$ and $B_r(r \times m)$ representing the reduced system and reduced distribution matrices respectively for the continuous case, and $\Phi_r(r \times r)$ and $\Gamma_r(r \times m)$ being the state transition matrix and distribution matrix for the reduced discrete case. Then the discretized reduced process can be described as

$$\underline{x}_r(k+1) = \Phi_r \underline{x}_r(k) + \Gamma_r \left[\underline{u}(k)+w(k)\right]$$

with measurement

$$z(k) = H\underline{x}(k) + v(k) \quad .$$

As in the partitioned method presented in Chapter 3, all the measured outputs are considered to be included in the reduced model, and furthermore the H matrix is assumed to be partitioned as

$$H = \left[H_1 \,\vdots\, H_2\right] \quad ,$$

with $H_2 = 0$, and $H_1(p \times r)$.

Considering the problem of initialization for the discrete reduced covariance equations, it is solved exactly as in Chapter 3, by partitioning the P(0/-1) matrix from the filter, as

$$P(0/-1) = \left[\begin{array}{c|c} P_{11}(0/-1) & P_{12}(0/-1) \\ \hline P_{21}(0/-1) & P_{22}(0/-1) \end{array}\right] ,$$

and choosing

$$P_r(0/-1) = P_{11}(0/-1)$$

with $P_{11}(0/-1)$ (r x r) and $P_r(0/-1)$ (r x r), the latter representing the initial value of covariance for the reduced filter.

For the reduced filter initial states,

$$\underline{\hat{x}}_r(0/-1) = \underline{\hat{x}}_1(0/-1)$$

where

$$\underline{\hat{x}}(0/-1) = \left[\begin{array}{c} \hat{x}_1(0/-1) \\ \hline \underline{\hat{x}}_2(0/-1) \end{array}\right]$$

B. EXAMPLE 3

The same example used in the previous methods is employed here to illustrate the method of dominant roots.

The same initialization values, same statistics for the random sequences, and same measurements as those used in Example 1, were used in this example for both the full and reduced systems. The corresponding values for the reduced filter are listed here for convenience

$$\underline{x}_r(0/-1) = \begin{bmatrix} 0 \\ \\ 0 \end{bmatrix}, \qquad P_r(0/-1) = \begin{bmatrix} 0.5 & 0 \\ \\ 0 & 2.0 \end{bmatrix},$$

$$H = \begin{bmatrix} 1 & 0 \end{bmatrix},$$

and

$$v(k) : N(0.0, 0.25),$$

$$w(k) : N(0.0, 0.01).$$

A significant difference between the two examples is the determination of the reduced transition and distribution matrices which in this case were computed as indicated by Eqs. (5.6) and (5.7), with

$$A_r = \begin{bmatrix} 0 & 1 \\ \\ -8 & -4 \end{bmatrix}, \qquad B_r = \begin{bmatrix} 0 \\ \\ 8 \end{bmatrix}$$

Due to the similarity between the partitioning method and the dominant-roots approach for this particular problem, the same computer program was used for the Monte Carlo simulation and solution of Examples 1 and 3, with the only change introduced regarding the computation of the dynamics for the reduced model.

Table (V-1) shows the initialization values for the reduced model, as well as the values for the Φ_r and Γ_r matrices. Tables (V-2) through (V-5) present numerical results of the simulation, and a graphical representation of the simulation is shown in Figs. (5.3) through (5.6).

Fig. (5.7) shows a complete block diagram of the reduced filtering process for this particular example.

Table V-1

PHI MATRIX

0.9999	0.0499	0.0012	0.0000
-0.0066	0.9946	0.0480	0.0009
-0.3644	-0.2981	0.8871	0.0307
-12.2713	-10.1815	-3.9181	0.3043

GAMMA MATRIX

0.0001
0.0066
0.3644
12.2713

REDUCED PHI MATRIX

0.9906	0.0452
-0.3613	0.8100

REDUCED GAMMA MATRIX

0.0093
0.3613

Q MATRIX(REDUCED)

0.0000	0.0000
0.0000	0.0000

VARIANCE OF FORCING NOISE = 0.01

VARIANCE OF MEASUREMENT NOISE = 0.25

REDUCED P(0/-1)

0.5000	0.0
0.0	2.0000

INITIALIZATION - X(0/-1)

0.0
0.0
0.0
0.0

Table V-2

GAINS FOR FULL AND REDUCED FILTERS

K	G(1,1)	G(2,1)	G(3,1)	G(4,1)	GR(1,1)	GR(2,1)
1	0.6667	0.0	0.0	0.0	0.6667	0.0
2	0.4071	0.2333	-0.2095	-7.2845	0.4014	0.0323
3	0.3097	0.4075	-0.6537	-9.4633	0.2896	0.0546
4	0.2677	0.5088	-1.1719	-9.0573	0.2286	0.0527
5	0.2477	0.5314	-1.6080	-7.0288	0.1895	0.0360
6	0.2349	0.4901	-1.8640	-4.4015	0.1613	0.0134
7	0.2227	0.4108	-1.9266	-1.9563	0.1390	-0.0091
8	0.2087	0.3175	-1.8378	-0.0721	0.1205	-0.0286
9	0.1928	0.2268	-1.6557	1.1851	0.1045	-0.0436
10	0.1758	0.1474	-1.4305	1.9077	0.0903	-0.0541
11	0.1584	0.0825	-1.1970	2.2348	0.0778	-0.0605
12	0.1412	0.0322	-0.9760	2.2961	0.0666	-0.0633
13	0.1247	-0.0051	-0.7780	2.1931	0.0566	-0.0633
14	0.1093	-0.0315	-0.6072	1.9984	0.0478	-0.0612
15	0.0950	-0.0490	-0.4637	1.7610	0.0401	-0.0576
16	0.0819	-0.0596	-0.3457	1.5120	0.0333	-0.0529
17	0.0701	-0.0647	-0.2505	1.2705	0.0274	-0.0477
18	0.0596	-0.0660	-0.1752	1.0474	0.0224	-0.0422
19	0.0502	-0.0643	-0.1167	0.8480	0.0181	-0.0367
20	0.0420	-0.0606	-0.0723	0.6746	0.0144	-0.0314
21	0.0348	-0.0556	-0.0395	0.5269	0.0114	-0.0264
22	0.0286	-0.0499	-0.0161	0.4036	0.0089	-0.0219
23	0.0233	-0.0438	-0.0001	0.3026	0.0069	-0.0179
24	0.0188	-0.0378	0.0100	0.2215	0.0052	-0.0143
25	0.0151	-0.0320	0.0157	0.1576	0.0039	-0.0113
26	0.0120	-0.0266	0.0181	0.1084	0.0029	-0.0088
27	0.0095	-0.0217	0.0181	0.0714	0.0021	-0.0067
28	0.0075	-0.0174	0.0166	0.0444	0.0016	-0.0050
29	0.0059	-0.0136	0.0141	0.0254	0.0011	-0.0037
30	0.0047	-0.0105	0.0111	0.0127	0.0008	-0.0026
31	0.0037	-0.0078	0.0080	0.0046	0.0006	-0.0018
32	0.0031	-0.0057	0.0050	-0.0000	0.0004	-0.0013
33	0.0026	-0.0041	0.0023	-0.0022	0.0003	-0.0008
34	0.0022	-0.0028	-0.0001	-0.0028	0.0003	-0.0006
35	0.0020	-0.0018	-0.0022	-0.0024	0.0002	-0.0004
36	0.0018	-0.0011	-0.0038	-0.0015	0.0002	-0.0002
37	0.0018	-0.0006	-0.0050	-0.0004	0.0002	-0.0002
38	0.0017	-0.0003	-0.0060	0.0007	0.0001	-0.0001
39	0.0017	-0.0001	-0.0066	0.0017	0.0001	-0.0001
40	0.0017	0.0000	-0.0070	0.0024	0.0001	-0.0001
41	0.0017	0.0001	-0.0073	0.0030	0.0001	-0.0001
42	0.0017	0.0001	-0.0074	0.0033	0.0001	-0.0001
43	0.0017	0.0001	-0.0074	0.0034	0.0001	-0.0001
44	0.0017	0.0001	-0.0073	0.0033	0.0001	-0.0001
45	0.0017	0.0000	-0.0072	0.0031	0.0001	-0.0001
46	0.0017	0.0000	-0.0071	0.0028	0.0001	-0.0001
47	0.0017	-0.0000	-0.0070	0.0025	0.0001	-0.0001
48	0.0017	-0.0000	-0.0068	0.0022	0.0001	-0.0001
49	0.0017	-0.0001	-0.0067	0.0018	0.0000	-0.0001
50	0.0017	-0.0001	-0.0066	0.0015	0.0000	-0.0001

G (4x1) represents the matrix of gains for the optimal filter.

GR(2x1) represents the matrix of gains for the reduced filter.

Table V-3

THEORETICAL VARIANCE OF ESTIMATION ERROR

K	PK(1,1)	PK(2,2)	PK(3,3)	PK(4,4)	PR(1,1)	PR(2,2)
1	0.1667	2.0000	2.0000	4.0000	0.1667	2.0000
2	0.1018	1.9603	1.7605	242.6279	0.1004	1.3335
3	0.0774	1.8002	2.3189	346.3362	0.0724	0.8822
4	0.0669	1.5182	3.9792	261.1479	0.0571	0.5793
5	0.0619	1.1771	5.7036	142.2309	0.0474	0.3794
6	0.0587	0.8478	6.6569	60.6997	0.0403	0.2498
7	0.0557	0.5745	6.6825	21.3209	0.0348	0.1672
8	0.0522	0.3709	6.0419	8.2223	0.0301	0.1153
9	0.0482	0.2307	5.0782	6.9855	0.0261	0.0832
10	0.0439	0.1400	4.0514	9.3412	0.0226	0.0636
11	0.0396	0.0847	3.1097	11.6978	0.0194	0.0516
12	0.0353	0.0529	2.3160	12.9672	0.0166	0.0442
13	0.0312	0.0361	1.6824	13.1242	0.0142	0.0394
14	0.0273	0.0282	1.1960	12.4778	0.0120	0.0360
15	0.0237	0.0255	0.8337	11.3650	0.0100	0.0332
16	0.0205	0.0254	0.5710	10.0516	0.0083	0.0308
17	0.0175	0.0264	0.3855	8.7171	0.0069	0.0285
18	0.0149	0.0275	0.2581	7.4696	0.0056	0.0261
19	0.0126	0.0283	0.1736	6.3652	0.0045	0.0237
20	0.0105	0.0286	0.1199	5.4260	0.0036	0.0212
21	0.0087	0.0282	0.0877	4.6526	0.0028	0.0188
22	0.0072	0.0273	0.0701	4.0336	0.0022	0.0165
23	0.0058	0.0259	0.0619	3.5512	0.0017	0.0142
24	0.0047	0.0241	0.0595	3.1854	0.0013	0.0121
25	0.0038	0.0221	0.0601	2.9159	0.0010	0.0102
26	0.0030	0.0199	0.0619	2.7237	0.0007	0.0084
27	0.0024	0.0177	0.0639	2.5918	0.0005	0.0069
28	0.0019	0.0155	0.0654	2.5057	0.0004	0.0056
29	0.0015	0.0133	0.0659	2.4529	0.0003	0.0044
30	0.0012	0.0114	0.0654	2.4236	0.0002	0.0035
31	0.0009	0.0096	0.0640	2.4099	0.0001	0.0027
32	0.0008	0.0081	0.0618	2.4059	0.0001	0.0021
33	0.0006	0.0067	0.0590	2.4071	0.0001	0.0016
34	0.0006	0.0056	0.0557	2.4105	0.0001	0.0012
35	0.0005	0.0046	0.0523	2.4141	0.0001	0.0008
36	0.0005	0.0039	0.0489	2.4167	0.0000	0.0006
37	0.0004	0.0033	0.0455	2.4177	0.0000	0.0004
38	0.0004	0.0028	0.0424	2.4170	0.0000	0.0003
39	0.0004	0.0024	0.0396	2.4146	0.0000	0.0002
40	0.0004	0.0022	0.0371	2.4109	0.0000	0.0002
41	0.0004	0.0020	0.0349	2.4062	0.0000	0.0001
42	0.0004	0.0018	0.0330	2.4008	0.0000	0.0001
43	0.0004	0.0017	0.0315	2.3951	0.0000	0.0001
44	0.0004	0.0017	0.0303	2.3894	0.0000	0.0001
45	0.0004	0.0017	0.0293	2.3839	0.0000	0.0001
46	0.0004	0.0016	0.0285	2.3787	0.0000	0.0001
47	0.0004	0.0016	0.0279	2.3741	0.0000	0.0001
48	0.0004	0.0016	0.0275	2.3700	0.0000	0.0001
49	0.0004	0.0016	0.0272	2.3664	0.0000	0.0000
50	0.0004	0.0016	0.0270	2.3635	0.0000	0.0000

PK represents the diagonal elements of the theoretical covariance matrix for the optimal filter.

PR represents the diagonal elements of the theoretical covariance matrix for the reduced filter.

Table V-4

EXPERIMENTAL VARIANCE OF ESTIMATION ERROR

K	VK(1,1)	VK(2,2)	VK(3,3)	VK(4,4)	VR(1,1)	VR(2,2)
1	0.1510	1.9066	1.9733	4.0880	0.1486	1.9041
2	0.1022	1.8632	1.7310	225.4341	0.1008	1.9624
3	0.0807	1.7518	2.2373	326.4941	0.0801	2.0096
4	0.0667	1.5112	3.8145	257.3462	0.0684	1.9437
5	0.0677	1.1743	5.6299	138.0874	0.0713	1.7625
6	0.0529	0.8372	6.0810	60.2167	0.0664	1.5128
7	0.0518	0.5928	6.5273	23.2761	0.0728	1.2314
8	0.0487	0.3939	6.0208	8.7271	0.0783	0.9501
9	0.0464	0.2513	5.1600	7.2510	0.0825	0.6993
10	0.0457	0.1612	4.4359	9.6855	0.0860	0.4897
11	0.0420	0.1002	3.5490	12.4134	0.0863	0.3264
12	0.0391	0.0617	2.7683	14.0120	0.0890	0.2020
13	0.0368	0.0401	2.0579	14.2776	0.0877	0.1218
14	0.0297	0.0300	1.3831	14.0637	0.0830	0.0696
15	0.0264	0.0256	0.9971	11.9529	0.0804	0.0388
16	0.0226	0.0247	0.6811	12.3539	0.0756	0.0270
17	0.0188	0.0260	0.4289	9.7735	0.0704	0.0274
18	0.0153	0.0261	0.2742	7.4905	0.0639	0.0335
19	0.0131	0.0271	0.1830	6.0835	0.0579	0.0440
20	0.0112	0.0269	0.1292	5.5882	0.0518	0.0552
21	0.0094	0.0274	0.0892	5.2477	0.0456	0.0661
22	0.0074	0.0252	0.0707	4.1038	0.0396	0.0745
23	0.0061	0.0240	0.0651	3.5547	0.0340	0.0811
24	0.0047	0.0216	0.0634	2.8311	0.0287	0.0846
25	0.0039	0.0202	0.0596	3.2233	0.0239	0.0858
26	0.0032	0.0185	0.0569	2.5204	0.0196	0.0844
27	0.0025	0.0166	0.0547	2.7723	0.0159	0.0809
28	0.0020	0.0149	0.0546	2.8033	0.0126	0.0756
29	0.0016	0.0130	0.0550	2.6636	0.0098	0.0688
30	0.0013	0.0114	0.0581	2.0651	0.0075	0.0612
31	0.0010	0.0098	0.0584	2.4213	0.0057	0.0532
32	0.0008	0.0083	0.0567	2.7746	0.0042	0.0453
33	0.0007	0.0071	0.0539	2.5252	0.0031	0.0379
34	0.0006	0.0060	0.0490	2.1855	0.0022	0.0313
35	0.0005	0.0050	0.0435	2.5705	0.0016	0.0254
36	0.0005	0.0042	0.0420	2.2536	0.0011	0.0204
37	0.0004	0.0035	0.0437	2.5342	0.0008	0.0162
38	0.0004	0.0029	0.0446	2.3070	0.0006	0.0127
39	0.0004	0.0025	0.0421	2.0512	0.0005	0.0098
40	0.0004	0.0022	0.0366	2.4148	0.0005	0.0075
41	0.0004	0.0019	0.0300	2.2901	0.0004	0.0056
42	0.0004	0.0018	0.0256	2.5146	0.0004	0.0043
43	0.0004	0.0017	0.0241	2.2052	0.0004	0.0033
44	0.0004	0.0016	0.0261	2.3978	0.0004	0.0025
45	0.0004	0.0015	0.0300	1.9640	0.0005	0.0020
46	0.0004	0.0014	0.0317	1.9171	0.0005	0.0017
47	0.0004	0.0014	0.0307	2.2013	0.0005	0.0015
48	0.0004	0.0014	0.0299	2.3214	0.0005	0.0015
49	0.0004	0.0015	0.0281	2.2047	0.0005	0.0015
50	0.0004	0.0016	0.0229	2.8952	0.0005	0.0016

VK represents the diagonal elements of the experimental covariance matrix for the optimal process.

VR represents the diagonal elements of the experimental covariance matrix for the reduced process.

Table V-5

EXPERIMENTAL MEANS OF ESTIMATION ERROR

K	MEANE1	MEANE2	MEANE3	MEANE4	MEANE1R	MEANE2R
1	0.0496	0.0500	-0.0350	-0.1543	0.0496	0.0500
2	0.0372	0.0392	-0.0572	-0.6276	0.0349	-0.1157
3	0.0356	0.0307	-0.0903	-0.7992	0.0266	-0.2315
4	0.0380	0.0271	-0.1340	-0.7676	0.0193	-0.3045
5	0.0395	0.0203	-0.1637	-0.3713	0.0100	-0.3441
6	0.0333	-0.0030	-0.1219	-0.0534	-0.0047	-0.3578
7	0.0299	-0.0146	-0.0909	0.2345	-0.0167	-0.3498
8	0.0044	-0.0565	0.1352	0.1478	-0.0403	-0.3249
9	-0.0017	-0.0536	0.1711	0.0767	-0.0515	-0.2894
10	0.0049	-0.0373	0.1013	0.1589	-0.0548	-0.2527
11	0.0142	-0.0265	0.0168	-0.0124	-0.0556	-0.2173
12	0.0021	-0.0281	0.0951	-0.0444	-0.0653	-0.1749
13	-0.0028	-0.0232	0.1136	-0.1564	-0.0704	-0.1356
14	-0.0072	-0.0168	0.1219	-0.3115	-0.0742	-0.0982
15	-0.0086	-0.0108	0.1078	-0.4010	-0.0757	-0.0651
16	-0.0058	-0.0083	0.0729	-0.3752	-0.0745	-0.0382
17	-0.0028	-0.0080	0.0511	0.0015	-0.0724	-0.0154
18	0.0020	-0.0112	0.0313	-0.0783	-0.0691	0.0028
19	0.0055	-0.0150	0.0153	-0.0996	-0.0657	0.0183
20	0.0044	-0.0137	0.0109	-0.1052	-0.0635	0.0337
21	0.0040	-0.0136	0.0142	0.2029	-0.0606	0.0456
22	0.0014	-0.0094	0.0236	0.1062	-0.0581	0.0570
23	0.0022	-0.0103	0.0273	0.0610	-0.0543	0.0640
24	0.0017	-0.0089	0.0262	-0.0824	-0.0506	0.0699
25	0.0011	-0.0072	0.0235	-0.0285	-0.0469	0.0738
26	0.0008	-0.0063	0.0215	-0.0504	-0.0430	0.0758
27	0.0007	-0.0056	0.0207	-0.0033	-0.0390	0.0762
28	0.0006	-0.0048	0.0263	0.1783	-0.0351	0.0755
29	0.0001	-0.0027	0.0329	0.1111	-0.0314	0.0744
30	0.0002	-0.0014	0.0292	-0.2192	-0.0277	0.0723
31	0.0003	-0.0006	0.0215	-0.1151	-0.0241	0.0693
32	0.0004	0.0002	0.0170	-0.0758	-0.0207	0.0656
33	0.0005	0.0008	0.0080	-0.2516	-0.0175	0.0613
34	0.0004	0.0011	0.0012	-0.0467	-0.0146	0.0564
35	0.0004	0.0012	0.0022	0.0639	-0.0119	0.0513
36	0.0005	0.0014	0.0016	-0.0675	-0.0094	0.0463
37	0.0006	0.0014	-0.0011	-0.0404	-0.0072	0.0413
38	0.0006	0.0012	-0.0069	-0.1685	-0.0053	0.0363
39	0.0006	0.0007	-0.0093	0.0404	-0.0036	0.0313
40	0.0006	0.0003	-0.0057	0.0877	-0.0022	0.0266
41	0.0007	0.0002	-0.0006	0.1207	-0.0009	0.0225
42	0.0007	0.0003	0.0027	0.0151	0.0001	0.0189
43	0.0008	0.0004	0.0038	0.0374	0.0010	0.0157
44	0.0008	0.0006	0.0026	-0.0628	0.0017	0.0129
45	0.0008	0.0007	0.0004	-0.0407	0.0022	0.0103
46	0.0008	0.0007	0.0035	0.1341	0.0027	0.0080
47	0.0007	0.0010	0.0075	0.0162	0.0031	0.0062
48	0.0008	0.0014	0.0069	-0.0321	0.0033	0.0048
49	0.0009	0.0017	0.0066	0.0174	0.0035	0.0036
50	0.0010	0.0020	0.0053	-0.0515	0.0037	0.0026

MEANE1 through MEANE4 represent the mean of estimation error in the complete state vector by means of optimal filtering.

MEANE1R and MEANE2R represent the mean of estimation error in states x_1 and x_2 by means of a reduced filter.

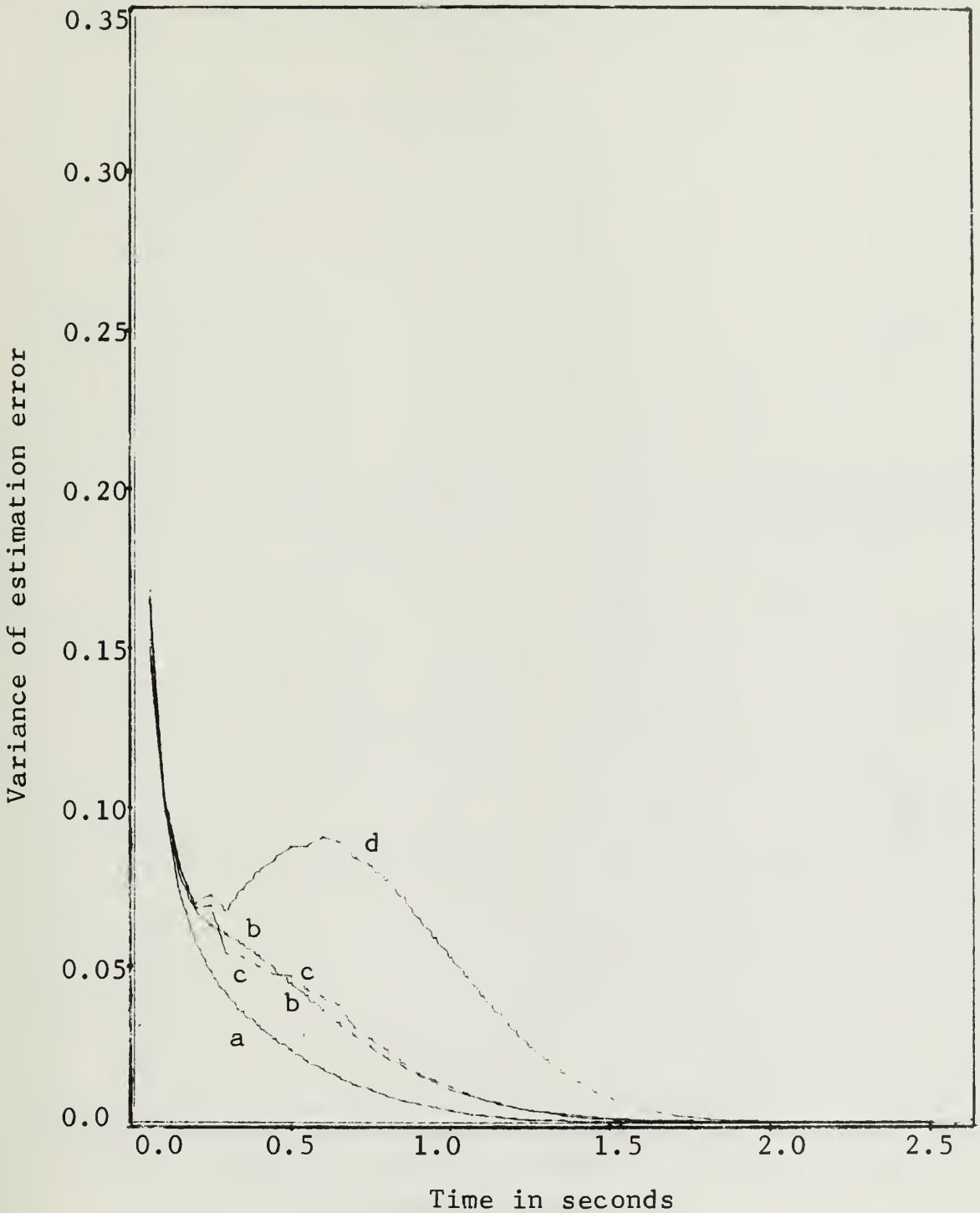


Fig. 5.3 - Graphical plot of the variance of estimation error in x_1 vs. time. Curves a and b show the theoretical variance of estimation error for the reduced and full filters respectively. Curves c and d show the experimental variance for the full and reduced processes respectively.

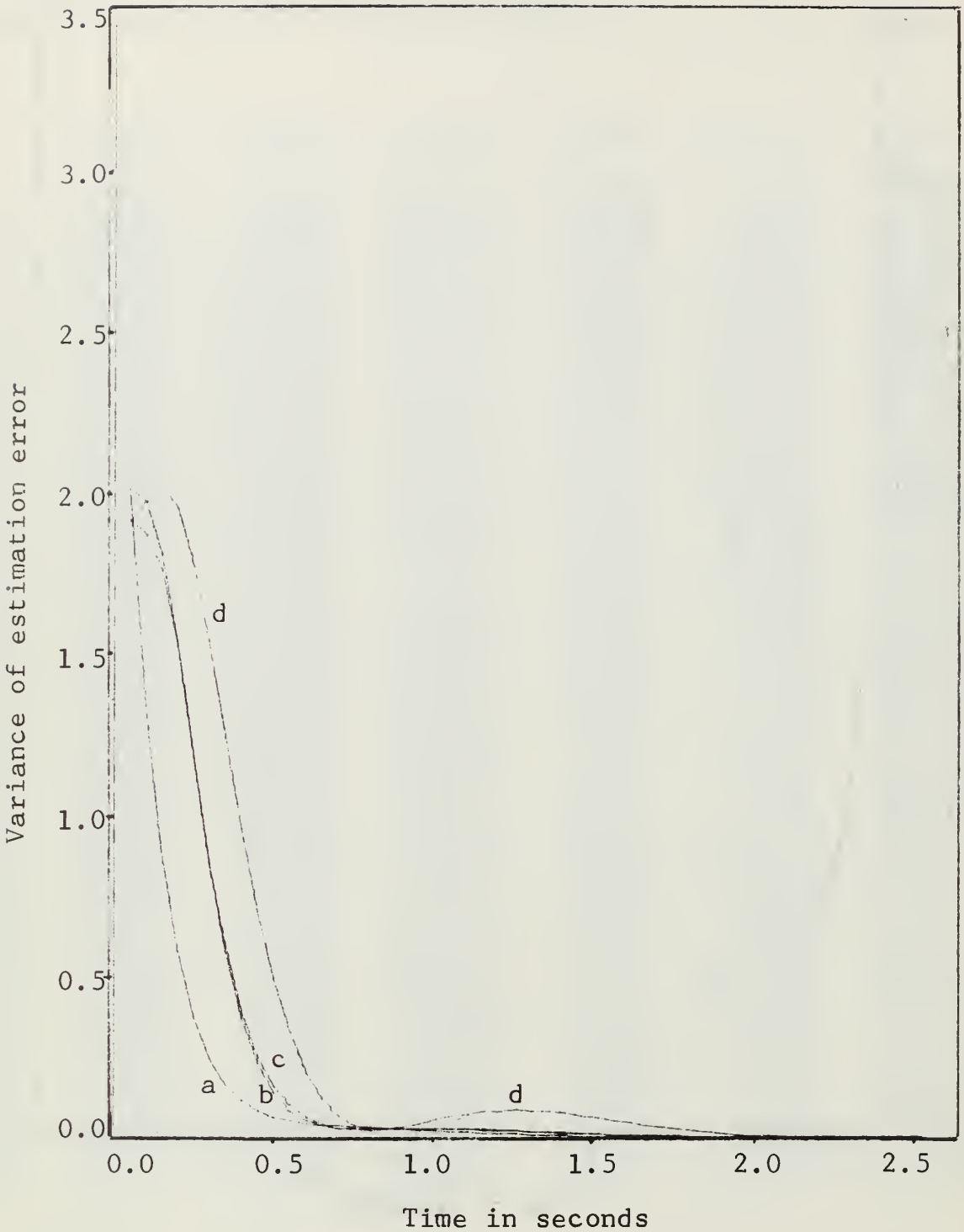


Fig. 5.4 - Graphical plot of the variance of estimation error in x_2 vs. time. Curves a and b show the theoretical variance of estimation error for the reduced and full filters respectively. Curves c and d show the experimental variance for the full and reduced processes respectively.

Fig. 5.5 - Graphical plot of the means of estimation error in x_1 vs. time. Curve a shows the mean of estimation error for the optimal Kalman filter, and curve b shows that for the reduced filter.

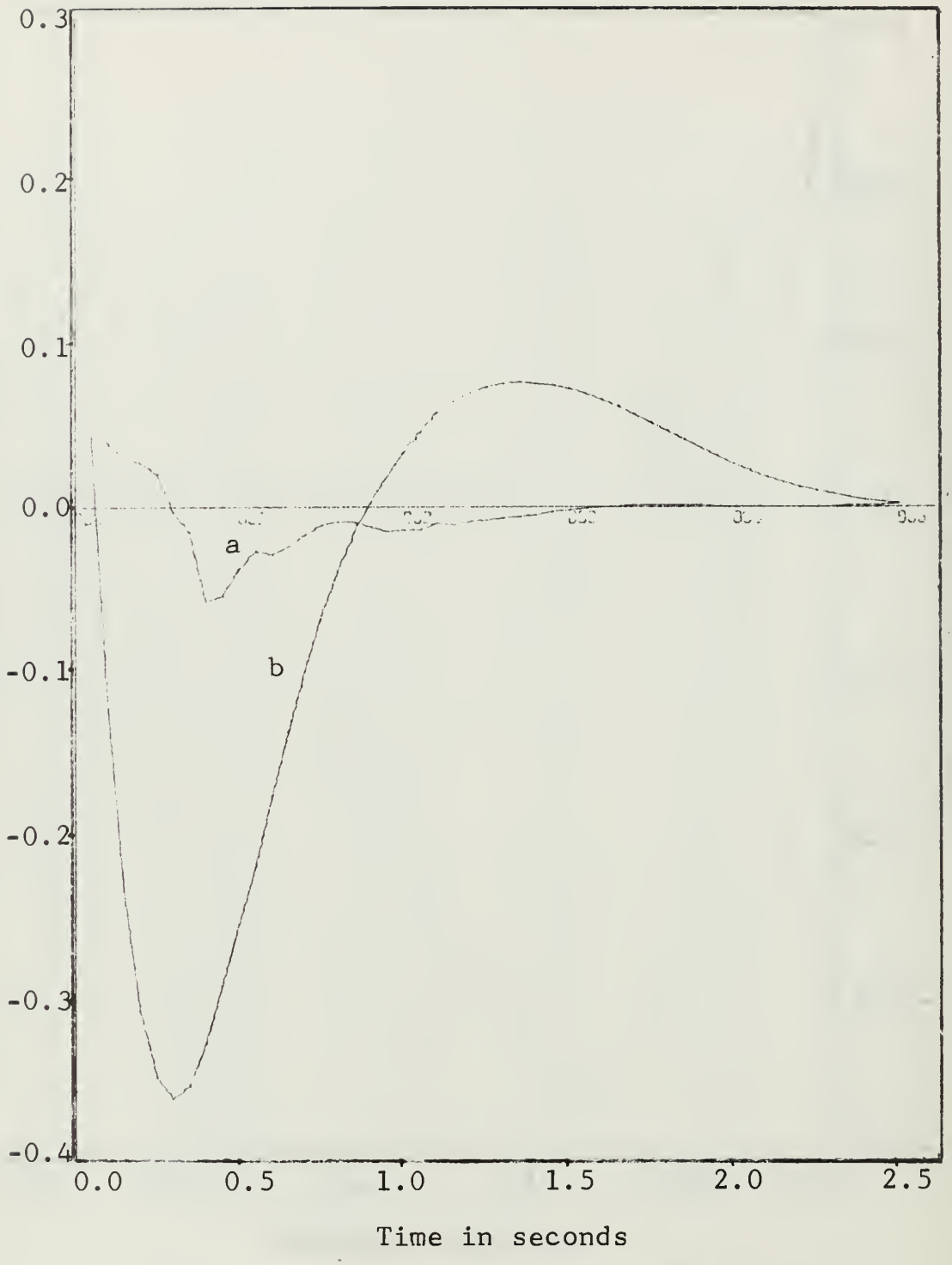


Fig. 5.6 - Graphical plot of the means of estimation error in x_2 vs. time. Curve a shows the mean of estimation error for the optimal Kalman filter, and curve b shows that for the reduced filter.

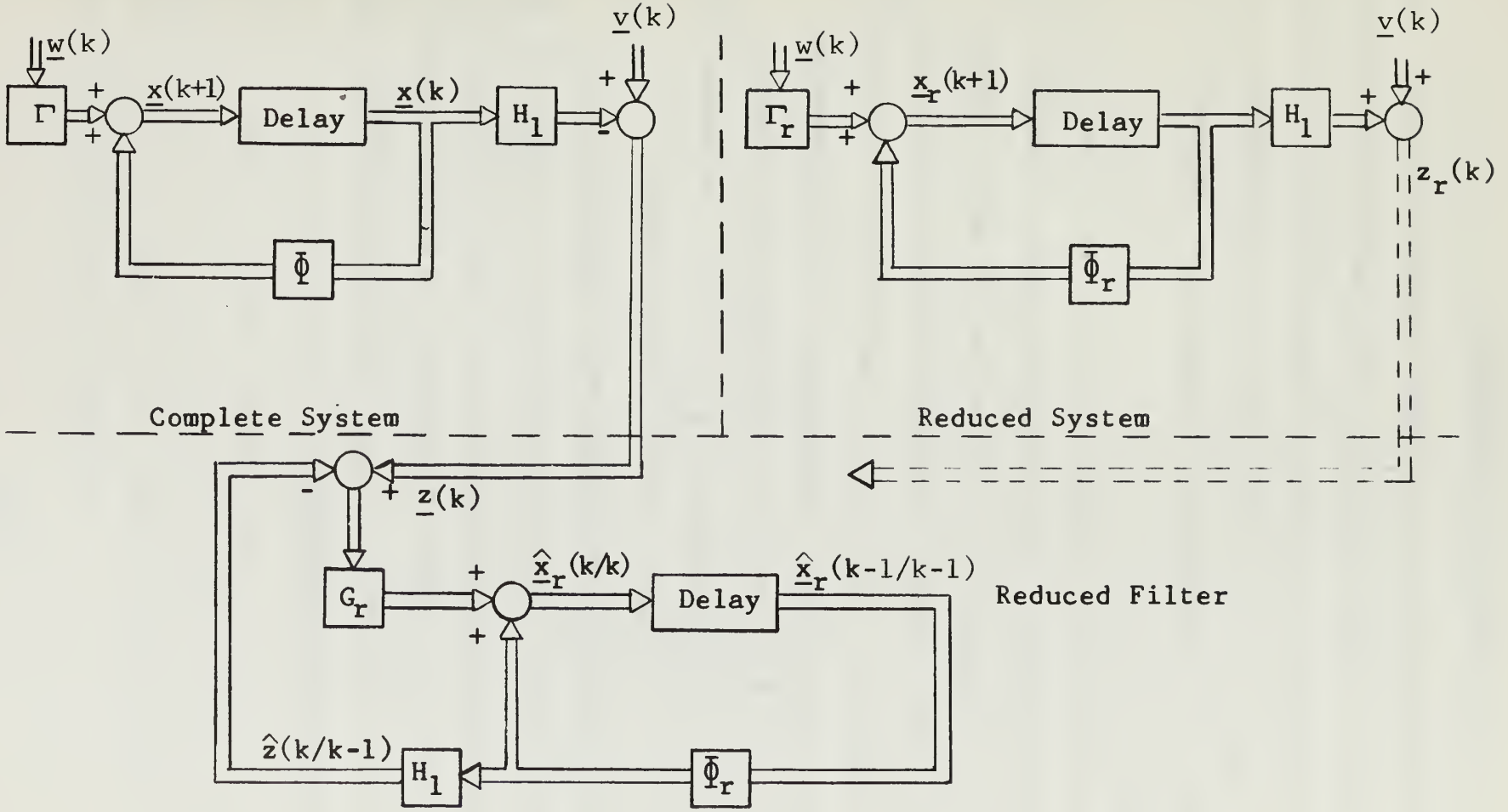


Fig. 5.7 - Block diagram showing the complete system and the reduced system and filter for the dominant-root implementation.

VI. DISCUSSION OF RESULTS

The three different approaches presented in this thesis have as a common objective a decrease in the computational requirements involved in handling systems of high dimension, based upon the fact that only a preselected reduced number (r) of states is of significant importance in the estimation process.

For the particular example chosen as an illustration for the three methods, the partitioning idea discussed in Chapter III, although it is the easiest to apply, appears to give the least attractive results in the reduced estimation of states x_1 and x_2 where a small bias is produced. This seems to be a direct consequence of the degradation introduced when Φ_{c1} (which in this example has relative large elements as compared with Φ_1) was not considered to be a part of the dynamics of the primary system.

In the pseudo-inversion approach, implementation of four scalar filters, plus the selection of a proper set of A_i matrices and its pseudo-inverses A_i^+ was necessary, but since it takes less computing capacity to manipulate four scalar filters than one (4 x 4) filter, the estimation process was clearly simplified.

The selection of the partitioning matrices A_i is arbitrary for any particular problem, therefore for this example another selection was made as

$$A_1 = \begin{bmatrix} 1 & 1 & -1 & -1 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 1 & -1 & -1 & 1 \end{bmatrix},$$

and

$$A_4 = \begin{bmatrix} 1 & -1 & 1 & -1 \end{bmatrix}$$

with

$$A_1^+ = \begin{bmatrix} 0.25 \\ 0.25 \\ -0.25 \\ -0.25 \end{bmatrix}, \quad A_2^+ = \begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{bmatrix}, \quad A_3^+ = \begin{bmatrix} 0.25 \\ -0.25 \\ -0.25 \\ 0.25 \end{bmatrix}, \quad A_4^+ = \begin{bmatrix} 0.25 \\ -0.25 \\ 0.25 \\ -0.25 \end{bmatrix}.$$

This particular choice, however, leads to four filters for which the sub-optimal variance of estimation error, both theoretical and experimental, increases linearly with time, which is undesireable. Therefore this particular partitioning was not used.

The plant for all three examples was selected with a pair of clearly dominant complex roots in order to obtain a good second-order approximation of the complete process from which the reduced filter could be implemented using the method presented in Chapter V.

APPENDIX A

MATRIX PSEUDO INVERSION

A review of matrix pseudo-inversion is presented in this appendix as an extension to the idea of pseudo-inversion introduced in Chapter IV.

A matrix A^{+} is said to be the pseudo-inverse of a rectangular matrix A if

$$A\,A^{+}A = A \tag{A-1}$$

where if A is a square non-singular matrix, i.e., A^{-1} exists, then

$$A\,A^{-1}A = A$$

is a trivial solution since $A\,A^{-1} = I$.

Another important definition may be introduced using a set of four matrix equations

$$A\,A^{+}\,A = A\ , \tag{A-2a}$$

$$A^{+}A\,A^{+} = A^{+}\ , \tag{A-2b}$$

$$\left[A\,A^{+}\right]^{T} = AA^{+}, \tag{A-2c}$$

and

$$\left[A^{+}A\right]^{T} = A^{+}A \tag{A-2d}$$

This definition implies that the inverse of a matrix A is established as the unique pseudo-inverse if and only if it satisfies the set of Eqs. (A-2). Furthermore, as a consequence of the latter definition, the pseudo-inverse among other properties has the following:

$$A^{++} = A$$

$$A^{T+} = A^{+T}$$

$$A^{+} = A^{-1}, \text{ when A is a non-singular matrix,}$$

and

$$0^{+} = 0^{T}.$$

The pseudo-inverse A^{+} of a matrix A may be shown to be unique as follows.

Let A_i^{+} be a set of pseudo-inverse of A_i where $i = 1,2 \ldots M$, and assume that the A_i^{+} are chosen so that

$$\sum_{i=1}^{M} A_i^{+} A_i = I \ . \qquad \text{(A-3)}$$

Further assume that another set of pseudo-inverse $Y_i^{+} \neq A_i^{+}$ exists for each A_i, and that this new set of Y_i^{+} also satisfies

$$\sum_{i=1}^{M} Y_i^{+} A_i = I \ . \qquad \text{(A-4)}$$

Then it follows by proper use of Eqs. (A-3) and (A-4) that

$$\sum_{i=1}^{M} (A_i^+ - Y_i^+)A_i = 0 \ ,$$

and since the choice of A_i is arbitrary, it follows that

$$Y_i^+ = A_i^+ \qquad \text{with} \quad i = 1,2 \ldots, M$$

and the uniqueness property is established.

For more complete treatment of this subject see Penrose (Ref.8), Werther (Ref.9) and Deutsch (Ref.10).

```
C       IMPLEMENTATION OF A REDUCED SUBOPTIMAL LINEAR FILTER
C       FOR DISCRETE DYNAMICS PROCESS, BY PARTITIONING THE
C       DYNAMICS OF THE ORIGINAL PROCESS. THE DEGRADATION ERR
C       OR OF THE FILTER IS COMPUTED AND USED AS A
C       MEANS OF CORRECTING THE FILTER VARIANCES. A MONTE CAR
C       LO SIMULATION OF THE PROCESS IS PERFORMED IN ORDER TO
C       OBTAIN APROXIMATED VALUES FOR THE MEANS AND COVARIANCE
C       OF ESTIMATION ERROR.
C
C
        DIMENSION GAM(4,1),GAMR(2,1),PHI(4,4),PHIR(2,2),A(4,4)
       1,H(1,4),B(4,1),HT(4,1),AT(4,4),GR(2,2),VW(1,1),Q(4,4)
       2,PK(4,4),HR(1,2),HTR(2,1),G(4,1),U(4,4),PK1R(2,2)
       3,PK1(4,4),PKR(2,2),VWR(1,1),GR(2,1),PKOR(2,2),COR(2,2)
       4,E(2,2),AX1(4,4),AX2(4,4),AX3(4,4),BX1(4,1),BX2(4,1)
       5,BX3(4,1),GAMU(4,1),DX2(1,4),DEG(2,2),DX3(2,2),UR(2,2)
       6,CX4(2,2),A1(2,2),A2(2,2),DX1(2,1),GAMUR(2,1),CX3(2,1)
       7,CX1(1,1),CX2(1,1),R(1,1),Y1(50,4),Y2(50,4),Y3(50,2)
       8,Y4(50,2),Y5(50,2),Y6(50,4),Y7(50,2),Y8(50,4),Y9(50,2)
       9,Y10(50,2),Y11(50,1),Y12(50,1),XE(4),XER(2),SE1(50),
       1SE2(50),SE3(50),SE4(50),SE1R(50),SE2R(50),SV11(50),
       2SV22(50),SV33(50),SV44(50),SVR11(50),SVR22(50),X(50),
       3LL(1),MM(1),LLL(1),MMM(1),PD1(50),PD2(50),VD1(50)
       4,VD2(50),PR1(50),PR2(50),VR1(50),VR2(50),ED1(50),
       5ED2(50),ER1(50),ER2(50)
        REAL*8 IT(12)/'VARIANCES OF ESTIMATION ERROR FOR X(1)
       1 LARA BOX11 PART. CORR. E=P                '/
        REAL*8 IP(12)/'VARIANCES OF ESTIMATION ERROR FOR X(2)
       1 LARA BOX11 PART. CORR. E=P                '/
        REAL*8 IE(12)/'MEAN OF ESTIMATION ERROR FOR X(1)
       1 LARA BOX11 PART. CORR. E=P                '/
        REAL*8 IF(12)/'MEAN OF ESTIMATION ERROR FOR X(2)
       1 LARA BOX11 PART. CORR. E=P                '/
        REAL LAB/4H    /
        DATA NT,NS,NE,M,MR,N,NR,NZ,IX,T/40,50,200,1,1,4,2,1,3,
       10.05/
        DATA A(1,1),A(1,2),A(1,3),A(1,4),A(2,1),A(2,2),A(2,3),
       2A(2,4),A(3,1),A(3,2),A(3,3),A(3,4),A(4,1),A(4,2),A(4,3
       3),A(4,4)/0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,-400.,-32
       40.,-118.,-19./
        DATA B(1,1),B(2,1),B(3,1),B(4,1)/0.,0.,0.,400./
        DATA H(1,1),H(1,2),H(1,3),H(1,4)/1.,0.,0.,0./
        DATA HT(1,1),HT(2,1),HT(3,1),HT(4,1)/1.,0.,0.,0./
        DATA VW(1,1),RR,VWR(1,1),UK/0.01,0.25,0.01,0.5/
        DATA SV1,SW1,AW1,KADA,LCOR/0.5,0.1,0.0,1,5/
        DATA S1,S2,S3,S4/0.707,1.41,1.41,2.0/
        DATA AM1,AM2,AM3,AM4/0.0,0.0,0.0,0.0/
        WRITE(6,222)
        DO 31 K=1,NS
        SV33(K)=0.0
        SV44(K)=0.0
        SVR22(K)=0.0
        SE3(K)=0.0
        SE4(K)=0.0
        SE2R(K)=0.0
        SV11(K)=0.0
        SV22(K)=0.0
        SVR11(K)=0.0
        SE1(K)=0.0
        SE2(K)=0.0
        SE1R(K)=0.0
 31     CONTINUE
C
C       MONTE CARLO SIMULATION
C
        DO 1 L=1,NE
        CALL GAUSS(IX,S1,AM1,X1)
        CALL GAUSS(IX,S2,AM2,X2)
        CALL GAUSS(IX,S3,AM3,X3)
        CALL GAUSS(IX,S4,AM4,X4)
        X1K=X1
        X2K=X2
```

```
      X3K=X3
      X4K=X4
C
C     GENERATING PHI AND GAMMA
C
      IF(L.GT.1)GO TO 6
      CALL UNIT(U,N)
      DO 49 I=1,NR
      DO 49 J=1,NR
49    UR(I,J)=U(I,J)
      CALL UNIT(PHI,N)
      CALL UNIT(AX1,N)
      CALL SMPY(B,T,GAM,N,M,0)
      CALL SMPY(B,T,BX2,N,M,0)
      CALL SMPY(A,T,AT,N,N,0)
      DO 66 I=1,NT
      XL=I
      RP=1.0/XL
      RP1=1.0/(XL+1.0)
      CALL GMPRD(AX1,AT,A,N,N,N)
      CALL SMPY(A,RP,AX1,N,N,0)
      CALL GMADD(PHI,AX1,PHI,N,N)
      CALL GMPRD(AT,BX2,B,N,N,M)
      CALL SMPY(B,RP1,BX2,N,M,0)
      CALL GMADD(GAM,BX2,GAM,N,M)
 66   CONTINUE
      WRITE(6,130)
      DO 40 I=1,N
40    WRITE(6,125)(PHI(I,J),J=1,N)
      WRITE(6,131)
      WRITE(6,125)GAM
C
C     GENERATING Q
C
      CALL GMPRD(GAM,VW,BX1,N,M,M)
      CALL GMTRA(GAM,DX2,N,M)
      CALL GMPRD(BX1,DX2,Q,N,M,N)
      WRITE(6,132)
      DO 81 I=1,N
81    WRITE(6,125)(Q(I,J),J=1,N)
      WRITE(6,135)
      WRITE(6,136)
 6    CONTINUE
      DO 123 I=1,N
      DO 123 J=1,N
      PK1(I,J)=0.0
123   CONTINUE
      PK1(3,3)=2.
      PK1(4,4)=4.
      PK1(1,1)=0.5
      PK1(2,2)=2.
      XE(1)=0.
      XE(2)=0.0
      XE(3)=0.0
      XE(4)=0.0
      XER(1)=0.0
      XER(2)=0.0
      DO 124 I=1,NR
      DO 124 J=1,NR
      PK1R(I,J)=0.0
      E(I,J)=0.0
124   CONTINUE
      PK1R(1,1)=0.5
      PK1R(2,2)=2.0
      E(1,1)=0.5
      E(2,2)=2.0
      IF(L.GT.1)GO TO 51
      WRITE(6,134)
      DO 142 I=1,N
142   WRITE(6,125)(PK1(I,J),J=1,N)
      WRITE(6,133)
      WRITE(6,125)XE
```

```
      WRITE(6,138)
      DO 143 I=1,NR
  143 WRITE(6,125)(E(I,J),J=1,NR)
51    CONTINUE
C
      DO 3 K=1,NS
C
      CALL GAUSS(IX,SV1,0.0,V1)
      CALL GAUSS(IX,SW1,AW1,W1)
      IF(K.EQ.1)GO TO 4
      X1K=PHI(1,1)*X1+PHI(1,2)*X2+PHI(1,3)*X3+PHI(1,4)*X4
     1+GAM(1,1)*(UK+W1)
      X2K=PHI(2,1)*X1+PHI(2,2)*X2+PHI(2,3)*X3+PHI(2,4)*X4
     2+GAM(2,1)*(UK+W1)
      X3K=PHI(3,1)*X1+PHI(3,2)*X2+PHI(3,3)*X3+PHI(3,4)*X4
     3+GAM(3,1)*(UK+W1)
      X4K=PHI(4,1)*X1+PHI(4,2)*X2+PHI(4,3)*X3+PHI(4,4)*X4
     4+GAM(4,1)*(UK+W1)
C
C     MEASUREMENT
C
  4   Z=H(1,1)*X1K+V1
C
C     GAIN AND COVARIANCES OF EST, ERROR FOR COMPLETE SYSTEM
C
      CALL GMPRD(PK1,HT,BX1,N,N,NZ)
      CALL GMPRD(H,BX1,CX2,NZ,N,NZ)
      SC=1.0/(CX2(1,1)+RR)
      CALL SMPY(BX1,SC,G,N,NZ,0)
      CALL GMPRD(G,H,AX1,N,NZ,N)
      CALL GMSUB(U,AX1,AX1,N,N)
      CALL GMPRD(AX1,PK1,PK,N,N,N)
      CALL GMTRA(PHI,AX2,N,N)
      CALL GMPRD(PK,AX2,AX1,N,N,N)
      CALL GMPRD(PHI,AX1,AX2,N,N,N)
      CALL GMADD(AX2,Q,PK1,N,N)
C
C     USING KALMAN FILTER EQUATION WITH FULL PLANT
C
      IF(K.GT.1)GO TO 7
      CALL GMPRD(U,XE,BX1,N,N,M)
      GO TO 8
7     CALL GMPRD(PHI,XE,BX1,N,N,M)
      CALL SMPY(GAM,UK,GAMU,N,M,0)
      CALL GMADD(BX1,GAMU,BX1,N,M)
8     CALL GMPRD(G,H,AX1,N,M,N)
      CALL GMSUB(U,AX1,AX2,N,N)
      CALL GMPRD(AX2,BX1,BX2,N,N,M)
      CALL SMPY(G,Z,BX1,N,M,0)
      CALL GMADD(BX1,BX2,XE,N,1)
      E1=X1K-XE(1)
      E2=X2K-XE(2)
      E3=X3K-XE(3)
      E4=X4K-XE(4)
C
C     GAIN AND COVAR. OF ESTIMATION ERROR FOR REDUCED SYSTEM
C
      IF(L.GT.1)GO TO 11
      DO 12 I=1,NR
      DO 12 J=1,NR
  12  PHIR(I,J)=PHI(I,J)
      DO 13 I=1,NR
      DO 13 J=1,MR
  13  GAMR(I,J)=GAM(I,J)
      DO 14 I=1,NR
      DO 14 J=1,NZ
      HR(J,I)=H(J,I)
  14  HTR(I,J)=HT(I,J)
C
C     Q REDUCED
C
      CALL GMPRD(GAMR,VWR,DX1,NR,MR,MR)
```

```
      CALL GMTRA(GAMR,CX2,NR,MR)
      CALL GMPRD(DX1,CX2,GR,NR,MR,NR)
      DO 44 I=1,NR
      DO 44 J=1,NR
 44   PKOR(I,J)=PK(I,J)
 11   CALL GMPRD(PK1R,HTR,DX1,NR,NR,NZ)
      CALL GMPRD(HR,DX1,CX2,NZ,NR,NZ)
      CX2(1,1)=1.0/(CX2(1,1)+RR)
      CALL GMPRD(DX1,CX2,GR,NR,NZ,NZ)
      CALL GMPRD(GR,HR,DX3,NR,NZ,NR)
      CALL GMSUB(UR,DX3,DX3,NR,NR)
      IF(LCOR.EQ.1)GO TO 1000
      IF(K.GT.1)GO TO 41
      CALL GMPRD(DX3,E,CCR,NR,NR,NR)
1000  CONTINUE
41    CALL GMPRD(DX3,PK1R,PKR,NR,NR,NR)
      CALL GMSUB(PKR,PKOR,DEG,NR,NR)
      CALL GMTRA(PHIR,DX4,NR,NR)
      IF(LCOR.EQ.1)GO TO 2000
      IF(K.EQ.1)GO TO 10
      CALL GMPRD(PHIR,E,A1,NR,NR,NR)
      CALL GMPRD(A1,DX4,A2,NR,NR,NR)
      CALL GMPRD(DX3,A2,CCR,NR,NR,NR)
10    DO 20 I=1,NR
      DO 20 J=1,NR
      PKR(I,J)=PKR(I,J)+CCR(I,J)
20    E(I,J)=CCR(I,J)
2000  CONTINUE
      CALL GMPRD(PKR,DX4,DX3,NR,NR,NR)
      CALL GMPRD(PHIR,DX3,DX4,NR,NR,NR)
      CALL GMADD(DX4,GR,PK1R,NR,NR)
C
C     USING KALMAN FILTER EQUATION WITH REDUCED PLANT
C
      IF(K.GT.1)GO TO 37
      CALL GMPRD(UR,XER,CX3,NR,NR,1)
      GO TO 17
 37   CALL GMPRD(PHIR,XER,CX1,NR,NR,1)
      CALL SMPY(GAMR,UK,GAMUR,NR,M,0)
      CALL GMADD(DX1,GAMUR,CX3,NR,1)
17    CALL GMPRD(GR,HR,DX3,NR,1,NR)
      CALL GMSUB(UR,DX3,DX4,NR,NR)
      CALL GMPRD(DX4,CX3,CX1,NR,NR,1)
      CALL SMPY(GR,Z,CX3,NR,NZ,0)
      CALL GMADD(CX3,DX1,XER,NR,1)
      E1R=X1K-XER(1)
      E2R=X2K-XER(2)
      X1=X1K
      X2=X2K
      X3=X3K
      X4=X4K
C
C     ENSEMBLE MEAN OF PK/K AND PK/K REDUCED, AND GAINS.
C
      IF(L.GT.1)GO TO 26
      Y1(K,1)=PK(1,1)
      Y2(K,1)=PK(2,2)
      Y3(K,1)=PK(3,3)
      Y1(K,3)=PKR(1,1)
      Y8(K,1)=G(1,1)
      Y8(K,2)=G(2,1)
      Y8(K,3)=G(3,1)
      Y8(K,4)=G(4,1)
      Y2(K,3)=PKR(2,2)
      Y9(K,1)=PK(4,4)
      Y10(K,1)=GR(1,1)
      Y10(K,2)=GR(2,1)
C
C     REAL VARIANCE OF EST. ERROR FOR FULL AND REDUCED SYST.
C
 26   SV11(K)=SV11(K)+E1**2
      SV22(K)=SV22(K)+E2**2
```

```
      SV33(K)=SV33(K)+E3**2
      SV44(K)=SV44(K)+E4**2
      SVR11(K)=SVR11(K)+E1R**2
   42 SVR22(K)=SVR22(K)+E2R**2
C
C     MEAN OF ESTIM. ERROR FOR FULL AND REDUCED PLANT
C
      SE1(K)=SE1(K)+E1
      SE2(K)=SE2(K)+E2
      SE3(K)=SE3(K)+E3
      SE4(K)=SE4(K)+E4
      SE1R(K)=SE1R(K)+E1R
      SE2R(K)=SE2R(K)+E2R
C
C     SYSTEM DEGRADATION
C
      IF(LCOR.EQ.1)GO TO 3000
      IF(L.GT.1)GO TO 27
      Y6(K,1)=DEG(1,1)
      Y6(K,2)=DEG(2,2)
      Y6(K,3)=COR(1,1)
      Y6(K,4)=COR(2,2)
 3000 CONTINUE
   27 IF(L.LT.NE)GO TO 94
      DO 88 K1=1,NS
      Y1(K1,2)=SV11(K1)/NE
      Y2(K1,2)=SV22(K1)/NE
      Y3(K1,2)=SV33(K1)/NE
      Y9(K1,2)=SV44(K1)/NE
      Y4(K1,1)=SE1(K1)/NE
      Y5(K1,1)=SE2(K1)/NE
      Y4(K1,2)=SE1R(K1)/NE
      Y5(K1,2)=SE2R(K1)/NE
      Y1(K1,4)=SVR11(K1)/NE-Y4(K1,2)**2
      Y2(K1,4)=SVR22(K1)/NE-Y5(K1,2)**2
      Y11(K1,1)=SE3(K1)/NE
      Y12(K1,1)=SE4(K1)/NE
   88 X(K1)=K1
   94 CONTINUE
    2 CONTINUE
    1 CONTINUE
      DO 1111 K=1,NS
      PD1(K)=Y1(K,1)
      PR1(K)=Y1(K,3)
      VD1(K)=Y1(K,2)
      VR1(K)=Y1(K,4)
      PD2(K)=Y2(K,1)
      PR2(K)=Y2(K,3)
      VD2(K)=Y2(K,2)
      VR2(K)=Y2(K,4)
      ED1(K)=Y4(K,1)
      ED2(K)=Y5(K,1)
      ER1(K)=Y4(K,2)
      ER2(K)=Y5(K,2)
 1111 CONTINUE
      WRITE(6,222)
      WRITE(6,160)
      WRITE(6,555)
      DO 300 K=1,NS,KADA
      WRITE(6,111)K,Y8(K,1),Y8(K,2),Y8(K,3),Y8(K,4),Y10(K,1)
     1,Y10(K,2)
  300 CONTINUE
      WRITE(6,222)
      WRITE(6,161)
      WRITE(6,666)
      DO 77 K=1,NS,KADA
      WRITE(6,111)K,Y1(K,1),Y2(K,1),Y3(K,1),Y9(K,1),Y1(K,3)
     1,Y2(K,3)
   77 CONTINUE
      WRITE(6,222)
      WRITE(6,162)
      WRITE(6,777)
```

```
      DO 888 K=1,NS,KADA
      WRITE(6,111)K,Y1(K,2),Y2(K,2),Y3(K,2),Y9(K,2),Y1(K,4)
     1,Y2(K,4)
888   CONTINUE
      WRITE(6,222)
      WRITE(6,163)
      WRITE(6,999)
      DO 757 K=1,NS,KADA
      WRITE(6,111)K,Y4(K,1),Y5(K,1),Y11(K,1),Y12(K,1),Y4(K,2
     1),Y5(K,2)
757   CONTINUE
      IF(LCOR.EQ.1)GO TO 4000
      WRITE(6,222)
      WRITE(6,164)
      WRITE(6,998)
      DO 778 K=1,NS,KADA
      WRITE(6,111)K,Y6(K,1),Y6(K,2),Y6(K,3),Y6(K,4)
778   CONTINUE
4000  CONTINUE
      IF(LCOR.NE.1)WRITE(6,444)
      IF(LCOR.EQ.1)WRITE(6,433)
      WRITE(6,222)
      CALL DRAW(50,X,PR1,1,0,LAB,IT,0,0,0,0,0,2,6,12,0,L)
      CALL DRAW(50,X,PD1,2,0,LAB,IT,0,0,0,0,0,2,6,12,0,L)
      CALL DRAW(50,X,VR1,2,0,LAB,IT,0,0,0,0,0,2,6,12,0,L)
      CALL DRAW(50,X,VD1,3,0,LAB,IT,0,0,0,0,0,2,6,12,0,L)
      CALL DRAW(50,X,PR2,1,0,LAB,IP,0,0,0,0,0,2,6,12,0,L)
      CALL DRAW(50,X,PD2,2,0,LAB,IP,0,0,0,0,0,2,6,12,0,L)
      CALL DRAW(50,X,VD2,2,0,LAB,IP,0,0,0,0,0,2,6,12,0,L)
      CALL DRAW(50,X,VR2,3,0,LAB,IP,0,0,0,0,0,2,6,12,0,L)
      CALL DRAW(50,X,ER1,1,0,LAB,IE,0,0,0,0,0,2,6,12,0,L)
      CALL DRAW(50,X,ED1,3,0,LAB,IE,0,0,0,0,0,2,6,12,0,L)
      CALL DRAW(50,X,ER2,1,0,LAB,IF,0,0,0,0,0,2,6,12,0,L)
      CALL DRAW(50,X,ED2,3,0,LAB,IF,0,0,0,0,0,2,6,12,0,L)
30    FORMAT(2I5)
 111  FORMAT(9X,I2,12F10.4/)
125   FORMAT(9X,4F10.4/)
 222  FORMAT(1H1)
 160  FORMAT(//9X,'GAINS FOR FULL AND REDUCED FILTERS'//)
 161  FORMAT(//9X,'THEORETICAL VARIANCE OF ESTIMATION ERROR'
     1//)
 162  FORMAT(//9X,'REAL VARIANCE OF ESTIMATION ERROR'//)
 163  FORMAT(//9X,'MEANS OF ESTIMATION ERROR'//)
 164  FORMAT(//9X,'PERFORMANCE DEGRADATION'//)
 138  FORMAT(///9X,'INITIALIZATION-E0/-1'/)
555   FORMAT(///9X,'K',5X,'G(1,1)',4X,'G(2,1)',4X,'G(3,1)'
     1,4X,'G(4,1)',3X,'GR(1,1)',3X,'GR(2,1)'//)
666   FORMAT(///9X,'K',4X,'PK(1,1)',3X,'PK(2,2)',3X,
     2'PK(3,3)',3X,'PK(4,4)',3X,'PR(1,1)',3X,'PR(2,2)'//)
777   FORMAT(///9X,'K',4X,'VK(1,1)',3X,'VK(2,2)',3X,
     2'VK(3,3)',3X,'VK(4,4)',3X,'VR(1,1)',3X,'VR(2,2)'//)
999   FORMAT(///9X,'K',5X,'MEANE1',4X,'MEANE2',4X,'MEANE3'
     1,4X,'MEANE4',3X,'MEANE1R',3X,'MEANE2R'//)
998   FORMAT(///9X,'K',5X,'D(1,1)',4X,'D(2,2)',4X,'E(1,1)'
     1,4X,'E(2,2)'//)
444   FORMAT(//9X,'CORRECTION DUE TO DEGRADATION APPLIED')
433   FORMAT(//9X,'NO CORRECTION FOR DEGRADATION' )
130   FORMAT(9X,'PHI MATRIX'/)
131   FORMAT(///9X,'GAMMA MATRIX')
133   FORMAT(///9X,'INITIALIZATION-X0/-1'/)
134   FORMAT(///9X,'INITIALIZATION-P0/-1'/)
135   FORMAT(///9X,'VARIANCE OF FORCING NOISE=0.01')
136   FORMAT(///9X,'VARIANCE OF MEASUREMENT NOISE=0.25')
132   FORMAT(///9X,'Q MATRIX'/)
      STOP
      END
```

```
      SUBROUTINE UNIT(U,N)
      DIMENSION U(4,4)
      DO 1 I=1,N
      DO 1 J=1,N
      U(I,J)=0.0
      IF(J.EQ.I)U(I,J)=1.0
    1 CONTINUE
      RETURN
      END

      SUBROUTINE GAUSS(IX,S,AM,V)
C
      A=0.0
      DO 50 I=1,12
      CALL RANDU(IX,IY,Y)
      IX=IY
   50 A=A+Y
      V=(A-6.0)*S+AM
      RETURN
      END

      SUBROUTINE RANDU(IX,IY,YFL)
C
      IY=IX*65539
      IF(IY)5,6,6
    5 IY=IY+2147483647+1
    6 YFL=IY
      YFL=YFL*.4656613E-9
      RETURN
      END

      SUBROUTINE LOC(I,J,IR,N,M,MS)
C
      IX=I
      JX=J
      IF(MS-1) 10,20,30
   10 IRX=N*(JX-1)+IX
      GO TO 36
   20 IF(IX-JX) 22,24,24
   22 IRX=IX+(JX*JX-JX)/2
      GO TO 36
   24 IRX=JX+(IX*IX-IX)/2
      GO TO 36
   30 IRX=0
      IF(IX-JX) 36,32,36
   32 IRX=IX
   36 IR=IRX
      RETURN
      END

      SUBROUTINE SMPY(A,C,R,N,M,MS)
      DIMENSION A(1),R(1)
      CALL LOC(N,M,IT,N,M,MS)
      DO 1 I=1,IT
    1 R(I)=A(I)*C
      RETURN
      END
```

```
      SUBROUTINE MINV(A,N,D,L,M)
      DIMENSION A(1),L(1),M(1)
      D=1.0
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
   10 IF( ABS(BIGA)- ABS(A(IJ))) 15,20,20
   15 BIGA=A(IJ)
      L(K)=I
      M(K)=J
   20 CONTINUE
      J=L(K)
      IF(J-K) 35,35,25
   25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI) =HOLD
   35 I=M(K)
      IF(I-K) 45,45,38
   38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
   40 A(JI) =HOLD
   45 IF(BIGA) 48,46,48
   46 D=0.0
      RETURN
   48 DO 55 I=1,N
      IF(I-K) 50,55,50
   50 IK=NK+I
      A(IK)=A(IK)/(-BIGA)
   55 CONTINUE
      DO 65 I=1,N
      IK=NK+I
      HOLD=A(IK)
      IJ=I-N
      DO 65 J=1,N
      IJ=IJ+N
      IF(I-K) 60,65,60
   60 IF(J-K) 62,65,62
   62 KJ=IJ-I+K
      A(IJ)=HOLD*A(KJ)+A(IJ)
   65 CONTINUE
      KJ=K-N
      DO 75 J=1,N
      KJ=KJ+N
      IF(J-K) 70,75,70
   70 A(KJ)=A(KJ)/BIGA
   75 CONTINUE
      D=D*BIGA
      A(KK)=1.0/BIGA
   80 CONTINUE
      K=N
  100 K=(K-1)
      IF(K) 150,150,105
```

```
  105 I=L(K)
      IF(I-K) 120,120,108
  108 JQ=N*(K-1)
      JR=N*(I-1)
      DO 110 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
  110 A(JI) =HOLD
  120 J=M(K)
      IF(J-K) 100,100,125
  125 KI=K-N
      DO 130 I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
  130 A(JI) =HOLD
      GO TO 100
  150 RETURN
      END


      SUBROUTINE GMTRA(A,R,N,M)
      DIMENSION A(1),R(1)
C
      IR=0
      DO 10 I=1,N
      IJ=I-N
      DO 10 J=1,M
      IJ=IJ+N
      IR=IR+1
   10 R(IR)=A(IJ)
      RETURN
      END


      SUBROUTINE GMPRD(A,B,R,N,M,L)
      DIMENSION A(1),B(1),R(1)
C
      IR=0
      IK=-M
      DO 10 K=1,L
      IK=IK+M
      DO 10 J=1,N
      IR=IR+1
      JI=J-N
      IB=IK
      R(IR)=0
      DO 10 I=1,M
      JI=JI+N
      IB=IB+1
   10 R(IR)=R(IR)+A(JI)*B(IB)
      RETURN
      END


      SUBROUTINE GMSUB(A,B,R,N,M)
      DIMENSION A(1),B(1),R(1)
      NM=N*M
      DO 10 I=1,NM
   10 R(I)=A(I)-B(I)
      RETURN
      END
```

```
      SUBROUTINE GMADD(A,B,R,N,M)
      DIMENSION A(1),B(1),R(1)
      NM=N*M
      DO 10 I=1,NM
   10 R(I)=A(I)+B(I)
      RETURN
      END
```

```
C       IMPLEMENTATION OF A REDUCED SUB-OPTIMAL FILTER USING
CC      MATRIX PSEUDO INVERSION. FOUR ONE DIMENSIONAL FILT-
CC      ERS ARE RUN IN PARALLEL IN ORDER TO ESTIMATE ONLY TWO
CC      STATES OF A FOURTH ORDER SYSTEM. A MONTE CARLO SIMULA
CC      TION  IS PERFORMED IN ORDER TO OBTAIN APPROXIMATED
CC      VALUES FOR THE MEANS AND  EXPERIMENTAL COVARIANCES
CC      OF ESTIMATION ERROR FOR THE COMPLETE AND REDUCED PRO-
CCCC    CESSES.
C
C
        DIMENSION PHI(4,4),GAM(4,1),A(4,4),B(4,1),Q(4,4),H(1,4
       1),HT(4,1),P(1,1),G(4,1),PK1(4,4),PK(4,4),VW(1,1),U(4,4
       2),VWR(1,1),AX1(4,4),AX2(4,4),AX3(4,4),AT(4,4),S(4,4),
       3S1(4,4),S2(4,4),BX1(4,1),BX2(4,1),BX3(4,1),BX4(4,1),BX
       45(4,1),AI1(4,1),AI2(4,1),AI3(4,1),AI4(4,1),AY1(4,1),AY
       52(4,1),AY3(4,1),AY4(4,1),AY5(4,1),GAMU(4,1),A1(1,4),A2
       6(1,4),A3(1,4),A4(1,4),B11(1,4),B12(1,4),B13(1,4),B14(1
       7,4),GH(1,4),DX2(1,4),CX2(1,1),CX3(1,1),SE1(5 ),SE2(5 )
       8,SE3(5 ),SE4(5 ),SE1R(5 ),SE2R(5 ),SV11(5 ),SV22(5 ),
       9SV33(5 ),SV44(5 ),SVR11(5 ),SVR22(5 ),XAXIS(5 ),XE(4),
       1LL(2),MM(2),LLL(1),MMM(1),Y1(5 ,4),Y2(5 ,4),Y3(5 ,3),
       2Y4(5 ,2),Y5(5 ,2),Y6(5 ,1),Y8(5 ,6),Y9(5 ,3),Y11(5 ,1)
       3,PD2(5 ),VD1(5 ),VD2(5 ),SD1(5 ),SD2(5 ),VR1(5 ),VR2(5
       4 ),ED1(5 ),ED2(5 ),ER1(5 ),ER2(5 ),PD1(5 ),X(5 )
        REAL*8 IT(12)/'VARIANCES OF ESTIMATION ERROR FOR X(1)
       1 LARA BOX 11                                        '/
        REAL*8 IP(12)/'VARIANCES OF ESTIMATION ERROR FOR X(2)
       1 LARA BOX 11                                        '/
        REAL*8 IE(12)/'MEAN OF ESTIMATION ERROR FOR X(1)
       1 LARA BOX 11                                        '/
        REAL*8 IF(12)/'MEAN OF ESTIMATION ERROR FOR X(2)
       1 LARA BOX 11                                        '/
        REAL LAB/4H    /
        DATA NT,NS,NE,M,MR,N,NR,NZ,IX,T/4 ,5 ,2 ,1,1,4,1,1,3,
       1 .( 5/
        DATA A(1,1),A(1,2),A(1,3),A(1,4),A(2,1),A(2,2),A(2,3),
       2A(2,4),A(3,1),A(3,2),A(3,3),A(3,4),A(4,1),A(4,2),A(4,3
       3),A(4,4)/ .,1., .,  .,  .,1.,  .,  .,  .,1.,-4  .,-32
       4 .,-118.,-19./
        DATA B(1,1),B(2,1),B(3,1),B(4,1)/ .,  .,  .,4   ./
        DATA H(1,1),H(1,2),H(1,3),H(1,4)/1.,  .,  .,  ./
        DATA HT(1,1),HT(2,1),HT(3,1),HT(4,1)/1.,  .,  .,  ./
        DATA VW(1,1),R(1,1),VWR(1,1),KADA/ .1, .25, .1,1/
        DATA UK,SV1,SW1,AW1/ .5, .5, .1, .  /
        DATA V1,V2,V3,V4,AM1,AM2,AM3,AM4/ .7,7,1.41,1.41,2.  ,
       1 .  ,  .  ,  .  ,  .  /
        DATA A1(1,1),A1(1,2),A1(1,3),A1(1,4)/1.,  .,  .,  ./
        DATA A2(1,1),A2(1,2),A2(1,3),A2(1,4)/ .,1.,  .,  ./
        DATA A3(1,1),A3(1,2),A3(1,3),A3(1,4)/ .,  .,1.,  ./
        DATA A4(1,1),A4(1,2),A4(1,3),A4(1,4)/ .,  .,  .,1./
        DATA AI1(1,1),AI1(2,1),AI1(3,1),AI1(4,1)/1.,  .,  .,  ./
        DATA AI2(1,1),AI2(2,1),AI2(3,1),AI2(4,1)/ .,1.,  .,  ./
        DATA AI3(1,1),AI3(2,1),AI3(3,1),AI3(4,1)/ .,  .,1.,  ./
        DATA AI4(1,1),AI4(2,1),AI4(3,1),AI4(4,1)/ .,  .,  .,1./
        WRITE(6,222)
        DO 31 K=1,NS
        SV33(K)= .
        SV44(K)= .
        SE3(K)= .
        SE4(K)= .
        SV11(K)= .
        SV22(K)= .
        SVR11(K)= .
        SVR22(K)= .
        SE1(K)= .
        SE2(K)= .
        SE1R(K)= .
        SE2R(K)= .
31      CONTINUE
C
CC      MONTE CARLO SIMULATION
C
```

```
      DO 1 L=1,NE
      CALL GAUSS(IX,V1,AM1,X1)
      CALL GAUSS(IX,V2,AM2,X2)
      CALL GAUSS(IX,V3,AM3,X3)
      CALL GAUSS(IX,V4,AM4,X4)
      X1K=X1
      X2K=X2
      X3K=X3
      X4K=X4
C
C     GENERATING PHI AND GAMMA
C
      IF(L.GT.1)GO TO 6
      CALL UNIT(U,N)
      UR=1.0
      CALL UNIT(PHI,N)
      CALL UNIT(AX1,N)
      CALL SMPY(B,T,GAM,N,M,0)
      CALL SMPY(B,T,BX2,N,M,0)
      CALL SMPY(A,T,AT,N,N,0)
      DO 66 I=1,NT
      XL=I
      RP=1.0/XL
      RP1=1.0/(XL+1.0)
      CALL GMPRD(AX1,AT,A,N,N,N)
      CALL SMPY(A,RP,AX1,N,N,0)
      CALL GMADD(PHI,AX1,PHI,N,N)
      CALL GMPRD(AT,BX2,B,N,N,M)
      CALL SMPY(B,RP1,BX2,N,M,0)
      CALL GMADD(GAM,BX2,GAM,N,M)
 66   CONTINUE
      WRITE(6,130)
      DO 4 I=1,N
4     WRITE(6,125)(PHI(I,J),J=1,N)
      WRITE(6,131)
      WRITE(6,125)GAM
C
C     GENERATING Q
C
      CALL GMPRD(GAM,VW,BX1,N,M,M)
      CALL GMTRA(GAM,DX2,N,M)
      CALL GMPRD(BX1,DX2,Q,N,M,N)
      WRITE(6,132)
      DO 81 I=1,N
81    WRITE(6,125)(Q(I,J),J=1,N)
      WRITE(6,135)
      WRITE(6,136)
6     CONTINUE
      DO 123 I=1,N
      DO 123 J=1,N
      PK1(I,J)=0.0
123   CONTINUE
      PK1(1,1)=0.5
      PK1(2,2)=2.0
      PK1(3,3)=2.0
      PK1(4,4)=4.0
      DO 54 I=1,N
      DO 54 J=1,N
 54   S(I,J)=PK1(I,J)
      XE(1)=0.0
      XE(2)=0.0
      XE(3)=0.0
      XE(4)=0.0
C
C     INITIALIZATION VALUES FOR REDUCED FILTERS.
C
      CALL GMPRD(A1,PK1,DX2,NR,N,N)
      CALL GMPRD(DX2,AI1,PK11,NR,N,NR)
      CALL GMPRD(A2,PK1,DX2,NR,N,N)
      CALL GMPRD(DX2,AI2,PK12,NR,N,NR)
      CALL GMPRD(A1,XE,YE1,NR,N,NR)
      CALL GMPRD(A2,XE,YE2,NR,N,NR)
```

```
      CALL GMPRD(A4,PK1,CX2,NR,N,N)
      CALL GMPRD(DX2,AI4,PK14,NR,N,NR)
      CALL GMPRD(A3,PK1,CX2,NR,N,N)
      CALL GMPRD(DX2,AI3,PK13,NR,N,NR)
      CALL GMPRD(A3,XE,YE3,NR,N,NR)
      CALL GMPRD(A4,XE,YE4,NR,N,NR)
      IF(L.GT.1)GO TO 65
      WRITE(6,134)
      DO 142 I=1,N
  142 WRITE(6,125)(PK1(I,J),J=1,N)
      WRITE(6,133)
      WRITE(6,125)XE
   65 CONTINUE
C
      DO 3 K=1,NS
C
      CALL GAUSS(IX,SV1,0.0,V1)
      CALL GAUSS(IX,SW1,AW1,W1)
      IF(K.EQ.1)GO TO 4
      X1K=PHI(1,1)*X1+PHI(1,2)*X2+PHI(1,3)*X3+PHI(1,4)*X4
     1+GAM(1,1)*(UK+W1)
      X2K=PHI(2,1)*X1+PHI(2,2)*X2+PHI(2,3)*X3+PHI(2,4)*X4
     2+GAM(2,1)*(UK+W1)
      X3K=PHI(3,1)*X1+PHI(3,2)*X2+PHI(3,3)*X3+PHI(3,4)*X4
     3+GAM(3,1)*(UK+W1)
      X4K=PHI(4,1)*X1+PHI(4,2)*X2+PHI(4,3)*X3+PHI(4,4)*X4
     4+GAM(4,1)*(UK+W1)
C
C     MEASUREMENT
C
   4  Z=H(1,1)*X1K+V1
C
C     GAIN AND COVARIANCES OF EST, ERROR FOR COMPLETE SYSTEM
C
      CALL GMPRD(PK1,HT,BX1,N,N,NZ)
      CALL GMPRD(H,BX1,CX2,NZ,N,NZ)
      CX2(1,1)=1.0/(CX2(1,1)+R(1,1))
      CALL GMPRD(BX1,CX2,G,N,NZ,NZ)
      CALL GMPRD(G,H,AX1,N,NZ,N)
      CALL GMSUB(U,AX1,AX1,N,N)
      CALL GMPRD(AX1,PK1,PK,N,N,N)
      CALL GMTRA(PHI,AX3,N,N)
      CALL GMPRD(PK,AX3,AX1,N,N,N)
      CALL GMPRD(PHI,AX1,AX2,N,N,N)
      CALL GMADD(AX2,Q,PK1,N,N)
C
C     USING KALMAN FILTER EQUATION WITH FULL PLANT
C
      IF(K.GT.1)GO TO 7
      CALL GMPRD(U,XE,BX1,N,N,M)
      GO TO 8
   7  CALL GMPRD(PHI,XE,BX1,N,N,1)
      CALL SMPY(GAM,UK,GAMU,N,M,0)
      CALL GMADD(BX1,GAMU,BX1,N,M)
   8  CALL GMPRD(G,H,AX1,N,M,N)
      CALL GMSUB(U,AX1,AX2,N,N)
      CALL GMPRD(AX2,BX1,BX2,N,N,M)
      CALL SMPY(G,Z,BX1,N,M,0)
      CALL GMADD(BX1,BX2,XE,N,1)
      E1=X1K-XE(1)
      E2=X2K-XE(2)
      E3=X3K-XE(3)
      E4=X4K-XE(4)
C
C     COMPUTING PHI,GAMMA AND H VALUES FOR REDUCED FILTERS.
C
      IF(L.GT.1.OR.K.GT.1)GO TO 61
      CALL GMPRD(A1,PHI,B11,NR,N,N)
      CALL GMPRD(B11,AI1,PHI1,NR,N,NR)
      CALL GMPRD(A1,GAM,GAM1,NR,N,NR)
      CALL GMPRD(H,AI1,H1,NR,N,NR)
      HT1=H1
```

```
      CALL GMPRD(A2,PHI,B12,NR,N,N)
      CALL GMPRD(B12,AI2,PHI2,NR,N,NR)
      CALL GMPRD(A2,GAM,GAM2,NR,N,NR)
      CALL GMPRD(H,AI2,H2,NR,N,NR)
      HT2=H2
      CALL GMPRD(A3,PHI,B13,NR,N,N)
      CALL GMPRD(B13,AI3,PHI3,NR,N,NR)
      CALL GMPRD(A3,GAM,GAM3,NR,N,NR)
      CALL GMPRD(H,AI3,H3,NR,N,NR)
      HT3=H3
      CALL GMPRD(A4,PHI,B14,NR,N,N)
      CALL GMPRD(B14,AI4,PHI4,NR,N,NR)
      CALL GMPRD(A4,GAM,GAM4,NR,N,NR)
      CALL GMPRD(H,AI4,H4,NR,N,NR)
      HT4=H4
C
C     Q REDUCED
C
      CALL GMPRD(GAM1,VWR,CX1,NR,MR,MR)
      CALL GMTRA(GAM1,CX2,NR,MR)
      CALL GMPRD(CX1,CX2,Q1,NR,MR,NR)
      CALL GMPRD(GAM2,VWR,CX1,NR,MR,MR)
      CALL GMTRA(GAM2,CX2,NR,MR)
      CALL GMPRD(CX1,CX2,Q2,NR,MR,NR)
      CALL GMPRD(GAM3,VWR,CX1,NR,MR,MR)
      CALL GMTRA(GAM3,CX2,NR,MR)
      CALL GMPRD(CX1,CX2,Q3,NR,MR,NR)
      CALL GMPRD(GAM4,VWR,CX1,NR,MR,MR)
      CALL GMTRA(GAM4,CX2,NR,MR)
      CALL GMPRD(CX1,CX2,Q4,NR,MR,NR)
 61   CONTINUE
C
C     SUB-OPTIMAL FILTERS
C
      CALL GMPRD(PK11,HT1,CX1,NR,NR,NZ)
      CALL GMPRD(H1,CX1,CX2,NZ,NR,NZ)
      CX2(1,1)=1.0/(CX2(1,1)+R(1,1))
      CALL GMPRD(CX1,CX2,G1,NR,NZ,NZ)
      CALL GMPRD(G1,H1,CX1,NR,NZ,NR)
      CALL GMSUB(UR,CX1,CX1,NR,NR)
      CALL GMPRD(CX1,PK11,PR1,NR,NR,NR)
      CALL GMTRA(PHI1,CX2,NR,NR)
      CALL GMPRD(PR1,CX2,CX1,NR,NR,NR)
      CALL GMPRD(PHI1,CX1,CX2,NR,NR,NR)
      CALL GMADD(CX2,Q1,PK11,NR,NR)
      IF(K.GT.1)GO TO 73
      YP1=YE1
      YP2=YE2
      YP3=YE3
      YP4=YE4
      GU1=GAM1*UK
      GU2=GAM2*UK
      GU3=GAM3*UK
      GU4=GAM4*UK
      IF(K.EQ.1)GO TO 37
 73   CONTINUE
      CALL GMPRD(AI1,YE1,AY1,N,NR,NR)
      CALL GMPRD(AI2,YE2,AY2,N,NR,NR)
      CALL GMPRD(AI3,YE3,AY3,N,NR,NR)
      CALL GMPRD(AI4,YE4,AY4,N,NR,NR)
      CALL GMADD(AY1,AY2,AY1,N,NR)
      CALL GMADD(AY3,AY4,AY3,N,NR)
      CALL GMADD(AY3,AY1,AY1,N,NR)
      CALL GMPRD(PHI,AY1,AY5,N,N,NR)
      CALL GMPRD(A1,AY5,YP1,NR,N,NR)
      CALL GMPRD(A2,AY5,YP2,NR,N,NR)
      CALL GMPRD(A3,AY5,YP3,NR,N,NR)
      CALL GMPRD(A4,AY5,YP4,NR,N,NR)
      YP1=YP1+GU1
      YP2=YP2+GU2
      YP3=YP3+GU3
      YP4=YP4+GU4
```

```
 37   CONTINUE
      CALL GMPRD(AI1,YP1,AY1,N,NR,NR)
      CALL GMPRD(AI2,YP2,AY2,N,NR,NR)
      CALL GMPRD(AI3,YP3,AY3,N,NR,NR)
      CALL GMPRD(AI4,YP4,AY4,N,NR,NR)
      CALL GMADD(AY1,AY2,AY1,N,NR)
      CALL GMADD(AY3,AY4,AY3,N,NR)
      CALL GMADD(AY1,AY3,AY2,N,NR)
C
CC    FILTER 1
C
      CALL GMPRD(G1,H,GH,NR,NR,N)
      CALL GMPRD(GH,AY2,GA,NR,N,NR)
      GZ=G1*Z
      YE1=GZ+YP1-GA
C
CC    FILTER 2
C
      CALL GMPRD(PK12,HT2,CX1,NR,NR,NZ)
      CALL GMPRD(H2,CX1,CX2,NZ,NR,NZ)
      CX2(1,1)=1./(CX2(1,1)+R(1,1))
      CALL GMPRD(CX1,CX2,G2,NR,NZ,NZ)
      CALL GMPRD(G2,H2,CX1,NR,NZ,NR)
      CALL GMSUB(UR,CX1,CX1,NR,NR)
      CALL GMPRD(CX1,PK12,PR2,NR,NR,NR)
      CALL GMTRA(PHI2,CX2,NR,NR)
      CALL GMPRD(PR2,CX2,CX1,NR,NR,NR)
      CALL GMPRD(PHI2,CX1,CX2,NR,NR,NR)
      CALL GMADD(CX2,Q2,PK12,NR,NR)
C
      CALL GMPRD(G2,H,GH,NR,NR,N)
      CALL GMPRD(GH,AY2,GA,NR,N,NR)
      GZ=G2*Z
      YE2=GZ+YP2-GA
C
CC    FILTER 3
C
      CALL GMPRD(PK13,HT3,CX1,NR,NR,NZ)
      CALL GMPRD(H3,CX1,CX2,NZ,NR,NZ)
      CX2(1,1)=1./(CX2(1,1)+R(1,1))
      CALL GMPRD(CX1,CX2,G3,NR,NZ,NZ)
      CALL GMPRD(G3,H3,CX1,NR,NZ,NR)
      CALL GMSUB(UR,CX1,CX1,NR,NR)
      CALL GMPRD(CX1,PK13,PR3,NR,NR,NR)
      CALL GMTRA(PHI3,CX2,NR,NR)
      CALL GMPRD(PR3,CX2,CX1,NR,NR,NR)
      CALL GMPRD(PHI3,CX1,CX2,NR,NR,NR)
      CALL GMADD(CX2,Q3,PK13,NR,NR)
C
      CALL GMPRD(G3,H,GH,NR,NR,N)
      CALL GMPRD(GH,AY2,GA,NR,N,NR)
      GZ=G3*Z
      YE3=GZ+YP3-GA
C
CC    FILTER 4
C
      CALL GMPRD(PK14,HT4,CX1,NR,NR,NZ)
      CALL GMPRD(H4,CX1,CX2,NZ,NR,NZ)
      CX2(1,1)=1./(CX2(1,1)+R(1,1))
      CALL GMPRD(CX1,CX2,G4,NR,NZ,NZ)
      CALL GMPRD(G4,H4,CX1,NR,NZ,NR)
      CALL GMSUB(UR,CX1,CX1,NR,NR)
      CALL GMPRD(CX1,PK14,PR4,NR,NR,NR)
      CX2(1,1)=PHI4
      CALL GMPRD(PR4,CX2,CX1,NR,NR,NR)
      CALL GMPRD(PHI4,CX1,CX2,NR,NR,NR)
      CALL GMADD(CX2,Q4,PK14,NR,NR)
C
      CALL GMPRD(G4,H,GH,NR,NR,N)
      CALL GMPRD(GH,AY2,GA,NR,N,NR)
      GZ=G4*Z
      YE4=GZ+YP4-GA
```

```
C
C       SUBOPTIMAL ESTIMATES
C
        CALL GMPRD(AI1,YE1,BX1,N,NR,NR)
        CALL GMPRD(AI2,YE2,BX2,N,NR,NR)
        CALL GMPRD(AI3,YE3,BX4,N,NR,NR)
        CALL GMPRD(AI4,YE4,BX5,N,NR,NR)
        CALL GMADD(BX4,BX5,BX4,N,NR)
        CALL GMADD(BX1,BX2,BX3,N,NR)
        CALL GMADD(BX4,BX3,BX3,N,NR)
        E1R=X1K-BX3(1,1)
        E2R=X2K-BX3(2,1)
        X1=X1K
        X2=X2K
        X3=X3K
        X4=X4K
C
C       CALCULATING THECRETICAL VARIANCE OF ESTIMATION ERROR
C
        CALL GMPRD(AI1,G1,BX1,N,NR,NR)
        CALL GMPRD(AI2,G2,BX2,N,NR,NR)
        CALL GMPRD(AI3,G3,BX3,N,NR,NR)
        CALL GMPRD(AI4,G4,BX4,N,NR,NR)
        CALL GMADD(BX3,BX4,BX5,N,NR)
        CALL GMADD(BX1,BX2,BX3,N,NR)
        CALL GMADD(BX3,BX5,BX3,N,NR)
        CALL GMTRA(BX3,DX2,N,NR)
        CALL GMPRD(BX3,H,AX1,N,NR,N)
        CALL GMSUB(U,AX1,AX2,N,N)
        CALL GMTRA(AX2,AX1,N,N)
        IF(K.EQ.1)GO TO 18
        CALL GMPRD(S,AX3,A,N,N,N)
        CALL GMPRD(PHI,A,S,N,N,N)
18      CALL GMPRD(AX2,S,A,N,N,N)
        CALL GMPRD(A,AX1,S,N,N,N)
        CALL GMPRD(AX2,Q,A,N,N,N)
        CALL GMPRD(A,AX1,S1,N,N,N)
        CALL GMPRD(BX3,R,BX2,N,NR,NR)
        CALL GMPRD(BX2,DX2,S2,N,NR,N)
        CALL GMADD(S,S1,S,N,N)
        CALL GMADD(S,S2,S,N,N)
C
C       ENSEMBLE MEAN OF PK/K, PK/K SUB-OPTIMAL AND GAINS
C
        IF(L.GT.1)GO TO 26
        Y1(K,1)=PK(1,1)
        Y2(K,1)=PK(2,2)
        Y3(K,1)=PK(3,3)
        Y9(K,1)=PK(4,4)
        Y8(K,1)=G(1,1)
        Y8(K,2)=G(2,1)
        Y8(K,3)=G(3,1)
        Y8(K,4)=G(4,1)
        Y8(K,5)=G1
        Y8(K,6)=G2
        Y1(K,4)=S(1,1)
        Y2(K,4)=S(2,2)
        Y3(K,2)=S(3,3)
        Y9(K,2)=S(4,4)
C
C       REAL COV. OF EST. ERROR FOR FULL AND REDUCED SYSTEMS
C
 26     SV11(K)=SV11(K)+E1**2
        SV22(K)=SV22(K)+E2**2
        SV33(K)=SV33(K)+E3**2
        SV44(K)=SV44(K)+E4**2
        SVR11(K)=SVR11(K)+E1R**2
        SVR22(K)=SVR22(K)+E2R**2
C
C       MEAN OF ESTIMATION ERROR FOR FULL AND REDUCED PLANTS.
C
        SE1(K)=SE1(K)+E1
```

```
      SE2(K)=SE2(K)+E2
      SE3(K)=SE3(K)+E3
      SE4(K)=SE4(K)+E4
      SE1R(K)=SE1R(K)+E1R
      SE2R(K)=SE2R(K)+E2R
      IF(L.LT.NE)GO TO 55
      DO 88 K1=1,NS
      Y1(K1,3)=SV11(K1)/NE
      Y2(K1,3)=SV22(K1)/NE
      Y3(K1,3)=SV33(K1)/NE
      Y9(K1,3)=SV44(K1)/NE
      Y4(K1,1)=SE1(K1)/NE
      Y5(K1,1)=SE2(K1)/NE
      Y6(K1,1)=SE3(K1)/NE
      Y11(K1,1)=SE4(K1)/NE
      Y4(K1,2)=SE1R(K1)/NE
      Y5(K1,2)=SE2R(K1)/NE
      Y1(K1,2)=SVR11(K1)/NE-Y4(K1,2)**2
      Y2(K1,2)=SVR22(K1)/NE-Y5(K1,2)**2
 88   XAXIS(K1)=K1
 55   CONTINUE
 3    CONTINUE
 1    CONTINUE
      WRITE(6,222)
      WRITE(6,160)
      WRITE(6,555)
      DO 3[illegible] K=1,NS,KADA
 3    WRITE(6,111)K,Y8(K,1),Y8(K,2),Y8(K,3),Y8(K,4),Y8(K,5),
     1Y8(K,6)
      WRITE(6,222)
      WRITE(6,161)
      WRITE(6,666)
      DO 77 K=1,NS,KADA
77    WRITE(6,111)K,Y1(K,1),Y2(K,1),Y3(K,1),Y9(K,1)
      WRITE(6,222)
      WRITE(6,162)
      WRITE(6,777)
      DO 888 K=1,NS,KADA
888   WRITE(6,111)K,Y1(K,3),Y2(K,3),Y3(K,3),Y9(K,3),Y1(K,2),
     1Y2(K,2)
      WRITE(6,222)
      WRITE(6,163)
      WRITE(6,999)
      DO 757 K=1,NS,KADA
757   WRITE(6,111)K,Y4(K,1),Y5(K,1),Y6(K,1),Y11(K,1),Y4(K,2)
     1,Y5(K,2)
      WRITE(6,222)
      WRITE(6,164)
      WRITE(6,525)
      DO 535 K=1,NS,KADA
535   WRITE(6,111)K,Y1(K,4),Y2(K,4),Y3(K,2),Y9(K,2)
C
C     GRAPHICAL OUTPUT
C
      DO 1[illegible]  K=1,NS
      PD1(K)=Y1(K,1)
      PD2(K)=Y2(K,1)
      VD1(K)=Y1(K,3)
      VD2(K)=Y2(K,3)
      SD1(K)=Y1(K,4)
      SD2(K)=Y2(K,4)
      VR1(K)=Y1(K,2)
      VR2(K)=Y2(K,2)
      ED1(K)=Y4(K,1)
      ED2(K)=Y5(K,1)
      ER1(K)=Y4(K,2)
      ER2(K)=Y5(K,2)
      X(K)=XAXIS(K)
1[illegible]  CONTINUE
      CALL DRAW(5[illegible],X,PD1,1,[illegible],LAB,IT, , , , , ,2,6,12, ,L)
      CALL DRAW(5[illegible],X,VD1,2,[illegible],LAB,IT, , , , , ,2,6,12, ,L)
      CALL DRAW(5[illegible],X,SD1,2,[illegible],LAB,IT, , , , , ,2,6,12, ,L)
```

```
      CALL DRAW(50,X,VR1,3,0,LAB,IT,0,0,0,0,0,2,6,12,0,L)
      CALL DRAW(50,X,PD2,1,0,LAB,IP,0,0,0,0,0,2,6,12,0,L)
      CALL DRAW(50,X,VD2,2,0,LAB,IP,0,0,0,0,0,2,6,12,0,L)
      CALL DRAW(50,X,SD2,2,0,LAB,IP,0,0,0,0,0,2,6,12,0,L)
      CALL DRAW(50,X,VR2,3,0,LAB,IP,0,0,0,0,0,2,6,12,0,L)
      CALL DRAW(50,XAXIS,ED1,1,0,LAB,IE,0,0,0,0,0,2,6,8,0,L)
      CALL DRAW(50,XAXIS,ER1,3,0,LAB,IE,0,0,0,0,0,2,6,8,0,L)
      CALL DRAW(50,XAXIS,ED2,1,0,LAB,IF,0,0,0,0,0,2,6,8,0,L)
      CALL DRAW(50,XAXIS,ER2,3,0,LAB,IF,0,0,0,0,0,2,6,8,0,L)
3     FORMAT(2I5)
 111  FORMAT(9X,I2,12F10.4/)
 125  FORMAT(9X,4F10.4/)
 222  FORMAT(1H1)
525   FORMAT(///9X,'K',4X,'SR(1,1)',3X,'SR(2,2)',3X,
     1'SR(3,3)',3X,'SR(4,4)'//)
 555  FORMAT(//9X,'K',5X,'G(1,1)',4X,'G(2,1)',4X,'G(3,1)',4X
     1,'G(4,1)',3X,'G1(1,1)',3X,'G2(1,1)'//)
 666  FORMAT(//9X,'K',4X,'PK(1,1)',3X,'PK(2,2)',3X,'PK(3,3)'
     1,3X,'PK(4,4)'//)
 777  FORMAT(//9X,'K',4X,'VK(1,1)',3X,'VK(2,2)',3X,'VK(3,3)'
     1,3X,'VK(4,4)',3X,'VR(1,1)',3X,'VR(2,2)'//)
999   FORMAT(///9X,'K',5X,'MEANE1',4X,'MEANE2',4X,'MEANE3'
     1,4X,'MEANE4',3X,'MEANE1R',3X,'MEANE2R'//)
130   FORMAT(9X,'PHI MATRIX'/)
 131  FORMAT(///9X,'GAMMA MATRIX'/)
132   FORMAT(///9X,'Q MATRIX'/)
134   FORMAT(///9X,'INITIALIZATION-P0/-1'/)
133   FORMAT(///9X,'INITIALIZATION-X0/-1'/)
135   FORMAT(///9X,'VARIANCE OF FORCING NOISE=0.01')
136   FORMAT(///9X,'VARIANCE OF MEASUREMENT NOISE=0.25')
 160  FORMAT(//9X,'GAINS FOR FULL AND REDUCED FILTERS'//)
 161  FORMAT(//9X,'THEORETICAL VARIANCE OF ESTIMATION ERROR'
     1,1X,'- FULL PLANT'//)
 162  FORMAT(//9X,'REAL VARIANCE OF ESTIMATION ERROR'//)
 163  FORMAT(//9X,'MEANS OF ESTIMATION ERROR'//)
 164  FORMAT(//9X,'SUBOPTIMAL VARIANCE OF ESTIMATION ERROR'
     1//)
      STOP
      END


      SUBROUTINE UNIT(U,N)
      DIMENSION U(4,4)
      DO 1 I=1,N
      DO 1 J=1,N
      U(I,J)=0.0
      IF(J.EQ.I)U(I,J)=1.0
  1   CONTINUE
      RETURN
      END


      SUBROUTINE GAUSS(IX,S,AM,V)
C
      A=0.0
      DO 50 I=1,12
      CALL RANDU(IX,IY,Y)
      IX=IY
 50   A=A+Y
      V=(A-6.0)*S+AM
      RETURN
      END


      SUBROUTINE RANDU(IX,IY,YFL)
C
      IY=IX*65539
      IF(IY)5,6,6
    5 IY=IY+2147483647+1
    6 YFL=IY
      YFL=YFL*.4656613E-9
      RETURN
```

```
   48 DO 55 I=1,N
      IF(I-K) 5[illegible],55,5[illegible]
    5[illegible] IK=NK+I
      A(IK)=A(IK)/(-BIGA)
   55 CONTINUE
      DO 65 I=1,N
      IK=NK+I
      HOLD=A(IK)
      IJ=I-N
      DO 65 J=1,N
      IJ=IJ+N
      IF(I-K) 6[illegible],65,6[illegible]
    6 IF(J-K) 62,65,62
   62 KJ=IJ-I+K
      A(IJ)=HOLD*A(KJ)+A(IJ)
   65 CONTINUE
      KJ=K-N
      DO 75 J=1,N
      KJ=KJ+N
      IF(J-K) 7[illegible],75,7[illegible]
    7 A(KJ)=A(KJ)/BIGA
   75 CONTINUE
      D=D*BIGA
      A(KK)=1.[illegible]/BIGA
   8[illegible] CONTINUE
      K=N
  1[illegible] K=(K-1)
      IF(K) 15[illegible],15[illegible],1[illegible]5
  1[illegible]5 I=L(K)
      IF(I-K) 12[illegible],12[illegible],1[illegible]8
  1[illegible]8 JQ=N*(K-1)
      JR=N*(I-1)
      DO 11[illegible] J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
  11[illegible] A(JI) =HOLD
  12[illegible] J=M(K)
      IF(J-K) 1[illegible],10[illegible],125
  125 KI=K-N
      DO 13[illegible] I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
  13[illegible] A(JI) =HOLD
      GO TO 1[illegible]
  15[illegible] RETURN
      END


      SUBROUTINE GMTRA(A,R,N,M)
      DIMENSION A(1),R(1)
C
      IR=[illegible]
      DO 1[illegible] I=1,N
      IJ=I-N
      DO 1[illegible] J=1,M
      IJ=IJ+N
      IR=IR+1
    1[illegible] R(IR)=A(IJ)
      RETURN
      END


      SUBROUTINE GMPRD(A,B,R,N,M,L)
      DIMENSION A(1),B(1),R(1)
C
      IR=[illegible]
      IK=-M
      DO 1[illegible] K=1,L
```

```
      END

      SUBROUTINE LOC(I,J,IR,N,M,MS)
C
      IX=I
      JX=J
      IF(MS-1) 10,20,30
   10 IRX=N*(JX-1)+IX
      GO TO 36
   20 IF(IX-JX) 22,24,24
   22 IRX=IX+(JX*JX-JX)/2
      GO TO 36
   24 IRX=JX+(IX*IX-IX)/2
      GO TO 36
   30 IRX=0
      IF(IX-JX) 36,32,36
   32 IRX=IX
   36 IR=IRX
      RETURN
      END

      SUBROUTINE SMPY(A,C,R,N,M,MS)
      DIMENSION A(1),R(1)
      CALL LOC(N,M,IT,N,M,MS)
      DO 1 I=1,IT
    1 R(I)=A(I)*C
      RETURN
      END

      SUBROUTINE MINV(A,N,D,L,M)
      DIMENSION A(1),L(1),M(1)
      D=1.0
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
   10 IF( ABS(BIGA)- ABS(A(IJ))) 15,20,20
   15 BIGA=A(IJ)
      L(K)=I
      M(K)=J
   20 CONTINUE
      J=L(K)
      IF(J-K) 35,35,25
   25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI) =HOLD
   35 I=M(K)
      IF(I-K) 45,45,38
   38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
   40 A(JI) =HOLD
   45 IF(BIGA) 48,46,48
   46 D=0.0
      RETURN
```

```
      IK=IK+M
      DO 1  J=1,N
      IR=IR+1
      JI=J-N
      IB=IK
      R(IR)=
      DO 1  I=1,M
      JI=JI+N
      IB=IB+1
    1 R(IR)=R(IR)+A(JI)*B(IB)
      RETURN
      END

      SUBROUTINE GMSUB(A,B,R,N,M)
      DIMENSION A(1),B(1),R(1)
      NM=N*M
      DO 1  I=1,NM
    1 R(I)=A(I)-B(I)
      RETURN
      END

      SUBROUTINE GMADD(A,B,R,N,M)
      DIMENSION A(1),B(1),R(1)
      NM=N*M
      DO 1  I=1,NM
    1 R(I)=A(I)+B(I)
      RETURN
      END
```

LIST OF REFERENCES

1. R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", J. Basic Eng. Trans., ASME, Series D, 82, 35-45 (1960).

2. R. E. Kalman and R. S. Bucy, "New Results in Linear Filtering and Prediction Theory", J. Basic Eng. Trans. ASME, Series D, 83, 95-108 (1961).

3. J. S. Meditch, "Suboptimal Linear Filtering for Continuous Dynamic Processes", Aerospace Corp. Tech. Rept., Report No. TOR-469(5107-35)-2, December 1964.

4. J. R. Huddle and D. A. Wismer, "Degradation of Linear Filter Performance Due to Modeling Error", IEEE Trans. Automatic Control (Short Papers). pp.421-423. August 1968.

5. M. Aoki and J. R. Huddle, "Estimation of the State Vector of a Linear Stochastic System with a Constrained Estimator". IEEE Trans. Automatic Control (Short Papers), Vol. AC-12, pp.432-433, August 1967.

6. R. C. K. Lee, "Optimal Estimation, Identification, and Control". Research Monograph No.28. The M.I.T. Press, Cambridge (1964).

7. J. S. Demetry, "Notes on Estimation Theory". Naval Postgraduate School, Monterey, California.

8. R. Penrose, "A Generalized Inverse for Matrices", Proc. Cambridge Phil. Soc. 51, pp.406-413 (1955).

9. M. H. F. Werther, "PhD. Dissertation on the Recursive Algorithm for the Best Approximate Solution of Linear Equations with Applications to System Identification and State Estimation". Naval Postgraduate School, Monterey, California, October 1969.

10. R. Deutsch, "Estimation Theory", Prentice Hall, Inc., 1965.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Dr. J. S. Demetry Electrical Engineering Department Naval Postgraduate School Monterey, California 93940	2
4.	Dr. D. E. Kirk Electrical Engineering Department Naval Postgraduate School Monterey, California 93940	1
5.	Dr. H. A. Titus Electrical Engineering Department Naval Postgraduate School Monterey, California 93940	1
6.	LCDR. Juan Francisco Lara Comandancia General de la Marina Avenida Wollmer, San Bernardino Caracas, Venezuela, S.A.	3
7.	Captain Hugo Gallipoli Moller Comandancia General de la Marina Avenida Wollmer, San Bernardino Caracas, Venezuela, S.A.	1
8.	Captain O. Ortega Jugo Comandancia General de la Marina Avenida Wollmer, San Bernardino Caracas, Venezuela, S.A.	1
9.	LCDR. J. E. D'Paola Comandancia General de la Marina Avenida Wollmer, San Bernardino Caracas, Venezuela, S.A.	1
10.	LCDR. H. Medina Arellano Comandancia General de la Marina Avenida Wollmer, San Bernardino Caracas, Venezuela, S.A.	1

		No. Copies
11.	Comandancia General de la Marina Estado Mayor Naval I Sección Avenida Wollmer, San Bernardino Caracas, Venezuela, S.A.	1
12.	Escuela Naval de Venezuela Escuela de Post-Grado Meseta de Mamo Venezuela, S.A.	1
13.	Universidad Central de Venezuela Facultad de Ingeniería Escuela de Ingeniería Eléctrica Ciudad Universitaria Caracas, Venezuela, S.A.	1

DOCUMENT CONTROL DATA - R & D

Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY (Corporate author)
Naval Postgraduate School
Monterey, California 93940

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP

3 REPORT TITLE
On the Implementation of Reduced Sub-Optimal Kalman Filters, for Discrete, Linear, Stochastic Processes with Time-Invariant Dynamics.

4 DESCRIPTIVE NOTES (Type of report and, inclusive dates)
Master's Thesis; December 1969

5 AUTHOR(S) (First name, middle initial, last name)
Juan F. Lara

6 REPORT DATE
December 1969

7a. TOTAL NO. OF PAGES
133

7b. NO. OF REFS
10

8a. CONTRACT OR GRANT NO.

b. PROJECT NO

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT
Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY
Naval Postgraduate School
Monterey, California 93940

13. ABSTRACT

Three different approaches to the problem of implementing a reduced-order, sub-optimal Kalman filter for a discrete, linear stochastic process, with time-invariant dynamics, are presented.

A first method, A, is based upon the partitioning of the system dynamics. A second method, B, is implemented using matrix pseudo-inversion and a third method, C, is based upon reduction of the original process to one of lower order using the dominant roots of the system. An expression for the performance degradation in method A is derived. In method B, expression for the sub-optimal estimation error, and sub-optimal variance of estimation error are derived.

The several methods are applied to a fourth-order process for illustration.

DD FORM 1 NOV 65 1473 (PAGE 1)
S/N 0101-807-6811

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Reduced Kalman Filter						
Sub-Optimal Kalman Filter						
Reduced Estimation of Stochastic Processes						